
Recomendador adaptativo y entrenador Pre-Flop para NLHE



Trabajo de Fin de Grado

Juan Carlos Marco González
Iván Martín Herrero

Dirigida por el Doctor
Manuel Núñez García

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Junio 2016

Documento maquetado con T_EX_S v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

Recomendador adaptativo y entrenador Pre-Flop para NLHE

Trabajo de Fin de Grado

**Juan Carlos Marco González
Iván Martín Herrero**

Dirigida por el Doctor

Manuel Núñez García

**Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid**

Junio 2016

Copyright © Juan Carlos Marco González y Iván Martín Herrero

*A todas aquellas personas
que disfruten jugando al póker,
y que quieran tener
una pequeña ventaja.*

*“El poker con límite es una ciencia;
pero el sin límite es un arte.
En el primero estás disparando a una diana;
en el segundo, la diana cobra vida
y te devuelve los disparos.”*
Jack Strauss

Agradecimientos

*Los ases son más grandes que la vida y
más grandiosos que las montañas.*

Mike Caro

Juan Carlos Marco González

A mi familia y amigos por apoyarme en todo momento para conseguir este objetivo.

En especial para:

Mi madre Teresa, por darme esta oportunidad de futuro entre muchas otras, por realizar el esfuerzo económico que ello ha conllevado, por insistir en la ejecución de ello y por luchar por tus hijos.

A mi padre Juan Carlos, aunque ya no estés entre nosotros, jamás te olvidaré. Me enseñaste lo dura que era la vida y por esa lección que me diste, he llegado hasta aquí. Gracias Papá.

Sara, por ser un apoyo constante, por tanto cariño y por siempre estar tan pendiente de mí.

Mario, por los momentos de hermanos que pasamos juntos ;)

Etoó, por ser tan fiel, cariñoso y querer estar siempre con nosotros.

Ángeles y Luís, por interesaros por mí y acogerme como uno mas en vuestra familia.

A mis amigos Álvaro, Elena, Gabi, Isma y Juanje, por los buenos y malos momentos que hemos pasado juntos.

A mis amigos, compañeros de FP y de carrera, María y Lin, por la ayuda mutua que nos hemos dado para conseguir las metas.

A la Formación Profesional, porque aún no estando lo valorada que debería estar, se adquieren unos altos conocimientos prácticos los cuales han hecho que esta carrera sea mucho mas fácil de sacar.

A mi tutor Manuel, por impartirme la asignatura mas entretenida de la carrera, por haber podido lograr la matrícula de honor en ella y por tutorizar este TFG.

A mi compañero Iván, por formar parte de este TFG.

Gracias a todos.

Iván Martín Herrero

Al fin, después de estos años escribo estas líneas, las cuáles no podrían haber sido escrito sin el apoyo de todas las personas que procedo a agradecer.

Gracias a mi familia por todo el apoyo todos estos años, por soportarme en las épocas de exámenes, que han sido muchas. Sobre todo a mis padres, a mis hermanos y a mis abuelos, que son los que más tiempo han tenido que aguantarme nervioso por los exámenes, mis enfados por los suspensos, etc. Y aunque están incluidos en el anterior mención especial a mis padres, porque más de un verano sin su apoyo, tras un mal año no podría ahora mismo estar escribiendo estás líneas porque no habría terminado. Siempre estáis ahí cuando os he necesitado, siempre para devolverme al buen camino. Gracias.

Gracias a Manuel, por darme la oportunidad de hacer este Trabajo de Fin de Grado sobre una de mis ideas surgida tras la asignatura de Herramientas Informáticas para los Juegos de Azar, ya que me ha permitido hacerlo como un pequeño proyecto personal de una idea propia. Gracias además por todo el apoyo y la ayuda con el trabajo, y por aguantarnos en las reuniones y resolver todas nuestras dudas sobre póker. Gracias a Juan Carlos por su entusiasmo en el trabajo, por aceptar los cambios de horarios y planificación pero, sobre todo, gracias por todo su trabajo en sacar adelante esta idea.

Gracias también al profesor Pablo Moreno Ger, ya que aunque para este trabajo de fin de grado, parezca que no haya contribuido, lo ha hecho, ya que en mi primer año de carrera fue el profesor de programación (Pascal) y consiguió hacerme ver que no me había equivocado de carrera, que lo que yo

quería era programar, construir, inventar, programas nuevos y tras haberlos ideado, ponerlos en marcha.

Gracias a la asociación ASCII de la facultad, ya que todos ellos, la gente que está o ha estado ha sido un gran apoyo siempre, apuntes, exámenes, clases y además un entorno de diversión. Aquel año de carrera que decidimos ir unos cuantos a la Campus Party de Valencia (la última que hubo en España) fue inolvidable. Nunca dejéis que caiga la asociación, porque mucha gente os necesita, como yo os necesité, para seguir y no dejarlo todo de lado.

Gracias a todos mis amigos, que siempre han estado apoyándome: Héctor, Ángel, Ulises, Carmen, etc. Porque aunque tenáis vuestras cosas que hacer si os he necesitado en algún momento, habéis estado dándome una sonrisa, una noche de fiesta para animar, una conversación a altas horas de la noche, un entrenamiento de volley, lo que ha sido necesario. Puede que me deje a alguien en la pequeña lista de antes, pero han sido muchos años y mucha gente, y de algunos hablaré después, pero los que no estéis y me hayáis apoyado también van para vosotros estas líneas.

Algunos de los que faltan en las líneas anteriores de amigos, es porque quiero agradecerse con líneas propias.

Gracias a Miguel, Miky, porque empezamos a la vez, y ningún año nos hemos separado, ni siquiera las prácticas que hemos tenido que hacer juntos. Todos esos años de clases, exámenes, prácticas se han pasado y aprobado mucho más fácil gracias a ti, siempre ayudando a los demás, apoyándoles, eso me incluye a mí. También por esas conversaciones, tras un mal examen, o cuando una asignatura se atascaba y había que buscar la forma de entenderla. Pero también por los buenos momentos tras exámenes, como las sangriadas de ASCII. Gracias Miky, porque esta carrera no hubiera sido igual de divertida sin ti.

Gracias a Julia, nos conocimos en una época de exámenes estudiando AM, cuanto nos costó aprobarla, y la sonrisa al ver la nota de aprobado no tuvo precio. Gracias por apoyarme todos estos años, algunas veces has tenido la fe en mí que yo no tenía o había olvidado. Siempre estás ahí cuando necesito alguien con quien hablar, y durante este tiempo han sido muchas veces las que te he necesitado y nunca has faltado a la cita. Además todas esas prácticas hechas juntos donde nuestra simbiosis era perfecta cuando uno no quería el otro le animaba, y así siempre las hemos sacado adelante. Gracias por todo, ahora te toca terminar a ti.

Gracias a Ángel, Archiw, que decir de ti, te conocí en ASCII jugando al

LOL, y ya desde ese momento fuiste incluso mi maestro en el juego, para mejorar, así como en la vida misma ya que me descubriste nuevos grupos de música, noches de juegos de mesa, nuevas cosas en mi vida que me hicieron crecer. Además del apoyo para la carrera, siempre me preguntabas que era el examen y si podías ayudarme, lo intentabas. También gracias por esos viajes, Barcelona, Málaga, Toledo, Torrevieja, Cantabria, Valencia, juntos hemos visto un montón de lugares nuevos para mí, y los viajes en coche inolvidables, conmigo de DJ nunca faltaba música para todos los gustos, aunque el volumen se bajara misteriosamente si la canción no era a gusto del conductor. Gracias Archiw, por lo anterior y porque siempre he tenido tu apoyo, y he sabido que estarías ahí a una llamada.

Gracias a Alfonso, mi profesor de baile preferido, gracias por esas tardes de baile que conseguían sacarme de la cabeza todas las tensiones que hubiera tenido ese día y hacer que terminara el día relajado, alegre y agotado. También gracias por esas escapadas de recreación histórica, porque es un mundo nuevo en el que si puedo seguiré, ya que me encantó. Gracias por aportar nuevas diversiones a mi vida y formas de relajarme que me han servido de mucho estos años para terminar.

Por último y no menos importante, gracias a Paola, no hace mucho que nos hemos conocido, apenas este último curso, pero tuvimos esa conexión que se tiene en el momento, y siempre has estado apoyándome con todo este último año tan estresante, aguantándome también, y ayudándome activamente, este Trabajo de Fin de Grado no tendría una pinta tan bonita sin tu ayuda en la edición con L^AT_EX, y tu buen ver de estilo. Gracias por todo este apoyo, esta ayuda y por haber aparecido en un año que si no hubiera estado, te hubiera necesitado.

Resumen

El póker en todas sus modalidades, tanto presenciales como online, lleva mucho tiempo siendo uno de los juegos de apuestas y azar más jugados del mundo. Dentro de los diferentes tipos de partidas de póker (Omaha, NLHE, PLHE, etc.) el más jugado es el NLHE, sobre el cual se va a desarrollar este trabajo de fin de grado; particularmente la versión online de la aplicación PokerStars. Existen herramientas para recoger información de estas partidas (PokerTracker, Holdem Manager), pero de la persona que los maneje depende saber el uso de los datos que nos dan estas herramientas, además de la información de las tablas de rangos de autores como Janda (Janda, 2013), o la clasificación de manos de Sklansky-Chubukov (Muñoz, 2009).

Con este trabajo queremos dar un apoyo a las personas que se inicien en el mundo del póker a tomar decisiones acertadas aunque no sean óptimas contra todos los jugadores. Algunos jugadores jugaran peor que la óptima y serán objetivos para ser “explotados” y así aumentar nuestro beneficio. Para saber cuáles son estos jugadores explotables, usaremos los datos que nos proporciona la herramienta PokerTracker 4.

A la hora de obtener los datos en el momento de la jugada hemos preparado un reconocedor de imágenes adaptado a PokerStars para recoger los datos que hay en la pantalla en el momento de nuestro turno y así decidir lo que debemos hacer. También se ha implementado un *entrenador* de jugadas para aprender qué hacer frente a unas jugadas, que utiliza unos datos introducidos por el usuario o generados aleatoriamente, y dentro de las aleatorias se pueden simplificar para jugadores noveles a dos de las situaciones más comunes: Open Raise y 3-bet.

Palabras clave: Póker online, Entrenador, PokerStars, NLHE, PokerTracker

Abstract

Poker in all of its variants (both online and offline), has been for a long time one of the most played gambling and chance games in the world. Within the different variants of poker games (Omaha, NLHE, PLHE, etc.) the most played is NLHE, the topic of this Graduate Thesis; particularly the game in the online application PokerStars. There are tools to compile information on these games (PokerTracker, Holdem Manager), but the player must know how to use the information provided by these tools, as well as the different range tables defined by authors such as Janda (Janda, 2013), or the Sklansky-Chubukov rankings (Muñoz, 2009).

With this project we want to provide support to new players, for them to take good decisions even if they do not use an optimal strategy. Some opponents will play worse than optimally; they will become targets to be “exploited”, and thus increase our benefit. To know who these exploitable opponents are, we use data provided by PokerTracker 4.

In order, to obtain the information of the current hand we have prepared an image recognition tool adapted to PokerStars that captures the information on screen when the user must decide their next play. We have also implemented a trainer to learn what to do against a series of hands, using data that can be input by the user or randomly generated, and among the latter, it can be simplified for amateur players to two of the most common situations: Open Raise and 3-bet.

Keywords: Online poker, Trainer, PokerStars, NLHE, PokerTracker

Índice

Agradecimientos	IX
Resumen	XIII
Abstract	XV
1. Introducción	1
1.1. Introducción	1
1.2. Objetivos	2
1.3. Conceptos	3
1.3.1. Fases	3
1.3.2. Posiciones	4
1.3.3. Acciones	4
1.3.4. Rangos	5
1.3.5. Otros conceptos	5
1.4. Modalidad de Juego	6
1.5. Marco de trabajo	6
2. Introduction	9
2.1. Introduction	9
2.2. Objectives	10
2.3. Concepts	11
2.3.1. Phases	11
2.3.2. Positions	11
2.3.3. Actions	12
2.3.4. Ranges	12
2.3.5. Other concepts	13
2.4. Game Modality	13
2.5. Frame of work	14
3. Proceso de desarrollo	15
3.1. Introducción	15

3.2. Metodología Scrum	16
3.3. Aplicación de Scrum al trabajo de fin de grado	16
3.4. Contribuciones de Juan Carlos Marco González	19
3.4.1. Servidor PokerStove	19
3.4.2. Maestro	20
3.4.3. Trainer	24
3.5. Contribuciones de Iván Martín Herrero	25
3.5.1. Idea inicial del proyecto	25
3.5.2. Gestión del proyecto	25
3.5.3. Realización y maquetación de la memoria	25
3.5.4. Reconocedor de imagen	26
4. Diseño, uso y resultados	33
4.1. Diseño del software	33
4.1.1. Flujo del Recomendador	33
4.1.2. Flujo del Entrenador	34
4.1.3. Flujo de la herramienta de rangos	34
4.1.4. Ruta de PokerStars	34
4.2. Uso del programa	38
4.2.1. Recomendador	39
4.2.2. Entrenador	41
4.2.3. Rangos	43
4.2.4. Ruta de PokerStars	46
4.3. Resultados	47
5. Conclusiones y líneas futuras	51
5.1. Conclusiones	51
5.2. Problemas encontrados en el desarrollo y sus soluciones	52
5.3. Líneas futuras	53
6. Conclusions and future lines of work	55
6.1. Conclusions	55
6.2. Problems found in development and solutions	56
6.3. Future lines	57
A. Test de usabilidad	59
Bibliografía	65

Índice de figuras

1.1. Logo de PokerTracker 4	3
1.2. Ventana de PokerStars	7
2.1. PokerTracker 4 logo	11
2.2. PokerStars window	14
3.1. Tablero de juego de PokerStars	26
3.2. Ficha de posición en PokerStars	28
3.3. As de Corazones	29
3.4. As de Diamantes	30
3.5. Apuesta de un jugador	31
3.6. Bote total de la mesa	32
4.1. Diagrama de flujo desde el menú principal	34
4.2. Diagrama de flujo del recomendador	35
4.3. Diagrama de flujo del entrenador	36
4.4. Diagrama de flujo del cambio de los rangos	37
4.5. Diagrama de flujo del seleccionador de ruta de PokerStars	37
4.6. Aspecto del menú principal	38
4.7. Aspecto del recomendador mientras espera a encontrar la mesa	39
4.8. Aspecto del recomendador mientras espera al turno del usuario	40
4.9. Aspecto del recomendador cuando llega el turno del usuario	40
4.10. Aspecto del recomendador tras calcular una recomendación	40
4.11. Aspecto del entrenador	41
4.12. Entrenador tras generar una jugada Open Range	42
4.13. Entrenador tras evaluar la opción elegida	43
4.14. Aspecto de la ventana de Rangos	44
4.15. Ejemplo de creación de un Open Range en UTG	45
4.16. Ejemplo de modificación del Opening Ranges_CO	45
4.17. Ejemplo de modificación de Opening Ranges_CO, seleccionando entre el 10 % y el 50 % y modificando jugadas	46
4.18. Ventana de selección de ruta	46

4.19. Tiempo medio empleado en cada tarea	48
4.20. Media de errores por tarea	48
4.21. Puntuación media de satisfacción según los criterios de evaluación del cuestionario final	49

Capítulo 1

Introducción

*La persona que inventó el poker fue
brillante; pero la persona que inventó las
fichas fue un genio.*

Julius Weintraub

RESUMEN: En este primer capítulo presentaremos los objetivos de este trabajo y los conceptos necesarios para entenderlo.

1.1. Introducción

El póker online ha cogido mucha fuerza en los últimos años debido a las grandes apuestas que las casas de póker online han hecho con anuncios y promociones en televisión e internet. Por ello cada vez hay más gente que juega online, y debido a la facilidad que dan estas salas ha crecido el número de jugadores que juegan al póker para *pasar un rato* pero sin preparación. A estos jugadores les llamaremos, de aquí en adelante, “jugadores recreacionales”.

A este tipo de jugadores será hacia los cuales está indicado el producto software de este trabajo de fin de grado. Nuestra intención es que les permita mejorar su juego sin necesidad de preparación o lecturas para las que no (necesariamente) disponen de tiempo.

La variante más jugada en estas salas online es el No Limit Hold’Em, NLHE a partir de ahora. Una mano en este juego se divide en varias fases: Pre-Flop, Flop, Turn y River (sección 1.3). En cada fase hay una sucesión de apuestas entre los jugadores en las cuales no se impone ningún límite en la cuantía de las apuestas, de ahí el apelativo No Limit. Existen otras variantes en las que hay sí límites en el tamaño de las apuestas, como por ejemplo Pot

Limit Hold’Em en las que el bote es la apuesta máxima que alguien puede realizar.

La variante de póker Hold’Em se basa en que a cada jugador se le reparten dos cartas en la primera fase (Pre-Flop). En las siguientes fases se muestran hasta cinco cartas sobre la mesa. Los jugadores forman su mano usando una combinación de sus cartas y las de la mesa, usando un total de cinco cartas. La primera ronda de apuestas empieza en la posición tras la ciega grande o “Big Blind” (BB), llamada “Under the gun” (UTG). Una vez que haya al menos dos jugadores que hayan igualado la apuesta, o todos se hayan retirado ante la apuesta de alguno de los jugadores, se pasa a la siguiente fase (si se ha igualado alguna apuesta), el “Flop”. En el Flop la primera carta del mazo se descarta y se muestran tres cartas sobre la mesa y comienza una nueva ronda de apuestas a partir de la posición ciega pequeña o “Small Blind” (SB) con el mismo fin que en la primera fase. La siguiente fase (si la hubiera) es el “Turn”, en la que se descarta la primera carta del mazo y se muestra una en la mesa, continuando la ronda de apuestas igual que en el Flop. Por último para el “River” se procede igual que en el Turn y al final, si se igualan al menos dos apuestas los jugadores involucrados descubren sus cartas para ver quién tiene mejor mano y gana el “Pot” o bote (PokerStars.es, 2016a).

El caso de Heads Up (en el cuál un jugador se enfrenta a otro) con límite de apuesta máxima ha sido resuelto recientemente por la Universidad de Alberta (Oskari Tammelin y Bowling, 2015) con un algoritmo exploratorio de estados. Ello es posible ya que para dos jugadores el espacio de estados es elevado pero factible. Pero en el momento que aumentamos el número de jugadores o no ponemos límite a la apuesta, el número de estados empieza a crecer desorbitadamente y no se puede resolver nada más que con aproximaciones y acotaciones para disminuir el número de estados.

Basándonos en lo anterior, y como no es posible resolver el juego de forma algorítmica, decidimos mejorar las posibilidades del usuario dándole más oportunidades y explotando a los rivales más débiles y defendiéndonos frente a los rivales mejores que nosotros. Para ello, tomamos como punto de partida las tablas de rangos dados por Janda en su libro “Applications of NLHE” (Janda, 2013), y aumentaremos o disminuirémos los rangos de estas tablas basándonos en como juegan nuestros rivales.

1.2. Objetivos

Uno de los objetivos de este trabajo de fin de grado es presentar un recomendador en directo de la mejor opción usando las tablas de Janda. Ello

ayudará a los jugadores recreacionales a jugar al póker de una manera mas eficiente, al menos desde la suposición de que todo el mundo juega con estas tablas. En realidad solo algunas personas las usan, jugadores profesionales por lo general e incluso éstos tienen sus propias tablas adaptadas a como les gusta jugar y que varían en función de contra quien jueguen. Estas tablas podrían incluso no ser óptimas contra jugadores recreacionales, ya que al no jugar óptimamente podemos “explotar” sus debilidades. Esta consideración nos lleva a la segunda parte del trabajo de fin de grado: un recomendador adaptativo dependiendo de contra quien juguemos cada mano.

Para llevar a cabo esta ampliación necesitamos datos sobre como juegan nuestros rivales, para poder clasificarlos entre recreacionales y habituales¹. Para esto usaremos la herramienta PokerTracker 4 (Max-Value-Software, 2016) (figura 2.1) que recoge las estadísticas en una Base de datos de la cual podemos sacar la información que necesitamos de cada jugador contra el que ya hayamos jugado y tengamos información. Como en cualquier experimento, necesitaremos un mínimo número de manos para que los datos sean fiables.



Figura 1.1: Logo de PokerTracker 4

Para ambos recomendadores necesitaremos conocer la situación de la mesa en el momento en que sea nuestro turno para saber qué acción realizar en función de nuestras cartas y los rangos, y lo que hayan hecho nuestros rivales. Para poder recoger esta información se ha implementado un reconocedor de imágenes que obtiene datos en el momento del turno como nuestras cartas en mano, la posición en la que estamos y las apuestas previas de los jugadores anteriores si las hubiera.

1.3. Conceptos

1.3.1. Fases

- **Pre-Flop:** Primera fase del juego en la que cada jugador solo dispone de sus cartas en la mano.

¹En inglés se usa el término *regular* que se traduce usualmente, de forma errónea, como jugadores “regulares”

- **Flop:** Segunda fase del juego en la que tras descartar una carta del mazo se sacan 3 cartas del mazo y se muestran en la mesa.
- **Turn:** Tercera fase del juego en la que tras descartar una carta del mazo se muestra una más en la mesa
- **River:** Cuarta fase del juego en la que tras descartar una carta del mazo se muestra una más en la mesa y al finalizar las apuesta y si se igualan las apuestas de al menos dos jugadores, los involucrados muestran las cartas haciendo alguna jugada seleccionando 5 de entre las 5 cartas comunes de la mesa y las dos de las respectivas manos.

1.3.2. Posiciones

- **Ciega pequeña o “Small Blind” (SB):** Es el primer jugador en hablar tras la fase de Pre-Flop; además al iniciar la mano hace una apuesta de la mitad de la ciega grande. Es el penúltimo jugador en hablar en la fase Pre-Flop justo antes de la ciega grande.
- **Ciega grande o “Big Blind” (BB):** Es el segundo jugador jugador en hablar tras la fase de Pre-Flop, además al iniciar la partida hace una apuesta igual a la ciega grande. Es el último jugador en hablar en la fase de Pre-Flop.
- **Under the Gun (UTG):** Es el primer jugador en hablar en la fase Pre-Flop, y el tercero asumiendo que las dos ciegas siguen en las manos, en las siguientes fases. Es la posición donde tendremos nuestras tablas con los rangos de apertura más pequeños ya que partimos sin saber nada de los demás jugadores.
- **Mid Position (MP):** Son las posiciones intermedias (para mesas de más de seis jugadores hay varios jugadores en MP) tras UTG. Aquí nuestro rango será más amplio que en UTG pero no mucho más amplio.
- **Cut Off (CO):** Es el jugador tras MP y ampliaremos aún más el rango de apertura ya que disponemos de más información de los jugadores anteriores.
- **Button o Dealer (BTN):** Es el jugador que habla antes de las ciegas. Tendrá un rango bastante más amplio que CO y similar al de SB.

1.3.3. Acciones

- **Tirarse o Fold:** Es la acción de no jugar una mano y “tirar” las cartas, no aceptando la apuesta de un jugador previo a nosotros. Haremos esta acción cuando no nos sean favorables las cartas para poder ganar.

- **Ver o Call:** Es la acción de igualar la apuesta de un jugador anterior a nosotros para continuar a la siguiente fase o, si es el River, ver quien tiene la mejor jugada.
- **Limp:** Es una acción especial de Call que es ver la ciega grande sin subir.
- **Subir o Raise:** Es la acción de apostar más que alguna de las apuestas anteriores, o si no ha habido ninguna, hacer una apertura de apuestas.

1.3.4. Rangos

Todos los rangos dependen de la posición en la que estemos.

- **Rango de apertura u Opening Range (OR):** Es el rango de jugadas en las que haremos Raise cuando nos llega la mano sin apuestas o con solo las ciegas y los jugadores que hayan igualado la ciega grande en Pre-Flop (limpers).
- **Rango de 3-Bet (3BET):** Es el rango de jugadas con el que volveremos a subir la apuesta de alguien que ha subido la apuesta inicial o a alguno de los otros jugadores.
- **Rango de Fold to 3-Bet(F3BET):** Es el rango de jugadas con el que haremos Fold a una apuesta de 3-BET.
- **Rango de 4-Bet (4BET):** Es el rango de jugadas con el que haremos Raise a una apuesta de 3-BET.
- **Rango de Fold to 4-Bet(F4BET):** Es el rango de jugadas con el que haremos Fold a una apuesta de 4-BET.
- **Cold Calling (CC):** Es el rango de hacer call frente a alguna de las apuestas sin haber apostado antes.

1.3.5. Otros conceptos

Valores de la base de datos de PokerTracker 4 que usaremos

- **VPIP:** Indica el dinero voluntariamente añadido al bote y señala en que porcentaje de casos un jugador pone dinero en el bote antes del “Flop”
- **“Pre-Flop Raise” (PFR):** Señala cuán a menudo sube un jugador en la primera ronda de apuestas.

Algunos de los conceptos anteriores de rangos también aparecerán como datos de PokerTracker y harán referencia al porcentaje con el que un jugador realiza esa jugada.

También estos dos conceptos tienen importancia para mirar los rangos dependiendo a qué rival nos enfrentemos y su posición en la mesa:

- **“In Position” (IP):** Estar en posición frente a otro jugador significa hablar después que el jugador, por ejemplo el jugador en BTN estará IP frente a UTG.
- **“Out Position” (OP):** Estar fuera de posición frente a otro jugador significa que hablaremos antes que nuestro rival, por ejemplo el jugador en UTG estará OP frente a BTN

1.4. Modalidad de Juego

Como hemos dicho antes existen muchas modalidades de póker. En este trabajo de fin de grado nos centraremos en la versión de NLHE en la etapa de “Pre-Flop” y nuestras herramienstas consideradas. Esta es la fase en que cada jugador, con solo sus cartas en la mano, debe decidir si apostar o no en función de sus cartas y las probabilidades de hacer jugada con las cartas que salgan más adelante.

Además hemos elegido el escenario de una mesa con 9 jugadores activos debido a que los jugadores recreacionales tienden a jugar mesas grandes de torneos en las que suele haber ese número de jugadores por mesa.

1.5. Marco de trabajo

Todos los algoritmos y funciones de este trabajo de fin de grado han sido desarrollados en el lenguaje de programación Java. El marco de trabajo está orientado a dar un apoyo a los jugadores recreacionales que jueguen al póker online. Para ello los usuarios disponen de un recomendador de jugadas en directo y de un entrenador personalizable para practicar las distintas situaciones para llegar a jugar de manera más autónoma, sin el programa de apoyo.

El recomendador también se puede personalizar. Se pueden cambiar los rangos desde las opciones para que, una vez se sepa cómo quiere jugar un jugador y sus rangos propios, pueda añadirlos, o si no se siente cómodo jugando alguna mano incluida en los rangos predefinidos, quitarlas.

Por último se han usado librerías externas como Tess4J (Nguyen, 2016) para reconocer los números de las apuestas de los jugadores en la mesa a partir de una imagen. Además hay algunas restricciones debidas a la dificultad y el tiempo de tener en cuenta todos los cambios en la interfaz de la sala PokerStars (PokerStars.es, 2016b). Estas restricciones son:

- El jugador debe seleccionar sentarse en todas las mesas que juegue en la parte inferior centrado.
- El jugador debe seleccionar en las opciones de usuario el tema de cartas de cuatro colores, con fondos de color y números en blanco.
- Ninguna ventana debe ocultar las esquinas de la aplicación de PokerStars (figura 2.2) ya que son necesarias para poder capturar la mesa de juego.



Figura 1.2: Ventana de PokerStars

Capítulo 2

Introduction

*The guy who invented poker was bright,
but the guy who invented the chip was a
genius.*

Julius Weintraub

RESUMEN: In this first chapter we will show the objectives of this project, and explain the basic concepts needed to understand it.

2.1. Introduction

Online poker has seen an increase in popularity in the last few years, due to the effort that online poker sites have put into advertisement, both in TV and on the Internet. There are increasing numbers of players, especially unprepared players who want to *pass the time*. We'll call them "recreational players" from now on.

These are the players that this software is geared towards. Its aim is to help these recreational players improve without the need for preparation or readings that they don't (necessarily) have time for.

No Limit Hold'Em, or NLHE, is the most popular poker variant in these sites. A hand in this game is divided in different phases: Pre-Flop, Flop, Turn and River (section 2.3). In each phase there is a succession of bets between players in which no limit is imposed to the amount of the bets, hence "No Limit". There are other variants that have limits in the amount of the bets, such as Pot Limit Hold'Em where the maximum amount per bet is the pot.

In Hold’Em poker two cards are dealt to each player in the first phase (Pre-Flop). During the next stages, up to five cards will be shown on the table. The players will form their hand with a combination of their cards in hand and those on the board, using a total of five cards. The first round of bets starts in the position after the big blind (BB), called Under the Gun (UTG). If at least two players call the bet, or everyone has folded, the game continues on to the next phase (only if the bet has been called), “Flop”. In Flop the first card of the deck is discarded and the next three cards of the deck are shown on the board, after which a new round of bets begins from the Small Blind (SB). The next phase (if it happens) is called “Turn”; again the first card of the deck is discarded and a new card is shown on the table from the deck, continuing the betting from Flop. The last phase is the “River”, which is just like Turn, and in the end if two or more call the highest bet, the players who call show their cards to see who wins the Pot.

The case of a “Heads Up” (a player facing only one other player) with maximum bet limit has been solved recently by the University of Alberta (Oskari Tammelin y Bowling, 2015) with a state exploring algorithm. This case is possible because for two players the space of states is big, but reasonable. But the moment we add players or refrain from limiting the bets, the number of states starts growing so fast that it only can be resolved with approximations and limits to reduce the number of states.

Based on the aforementioned, and how it is not possible to resolve the game in an algorithmic way, we chose to improve the users’ chance, exploiting the weakest rivals and deferring from better players than us. For that, we start from Janda’s range tables from his book “Applications of NLHE” (Janda, 2013), and we open or close the ranges from these tables according to how our opponents plays.

2.2. Objectives

One of the objectives of this Graduate Thesis is to provide a live recommender for the best option using the Janda tables. This will help recreational players play poker in an efficient way, at least from the supposition that everybody plays using these tables. In fact, only some people use these tables, usually professional players, and even then, they will have their own tables configured to how they want to play and against whom. The Janda tables might even be non-optimal against recreational players, since their own non-optimal play allows us to “exploit” their weaknesses. This consideration takes us to the second part of this Graduate thesis: an adaptive recommender that changes depending on our rivals.

For this ampliation we need data about how our rivals play, to classify them between recreational and regular players. To this end we will use PokerTracker4 (Max-Value-Software, 2016) (image 2.1), a tool that provides a database of play statistics for players we've already met at the table. Like any experiment, we'll need a certain minimum number of hands to ensure data reliability.



Figura 2.1: PokerTracker 4 logo

For both recommenders we need to know the situation of the table on our turn to know what action to do according to our cards and ranges, and the actions taken by our rivals. To store that information an image recogniser will be developed and implemented that obtains data such as our cards in hand, the position that we are in and the previous bets of the players if they exist.

2.3. Concepts

2.3.1. Phases

- **Pre-Flop:** First phase of the game during which each player only has the cards in their hand.
- **Flop:** Second phase of the game during which, after discarding a card from the deck, another three are shown on the table.
- **Turn:** Third phase of the game during which, after discarding a card from the deck, one is shown on the table.
- **River:** Fourth phase of the game during which a card is discarded before showing one more on the table. Once the bets are finished and if at least two players call the bet, the players involved show their cards, that can make any of the poker hands with help from the five common cards.

2.3.2. Positions

- **Small Blind (SB):** First player to talk after the Pre-Flop phase; furthermore at the start of the hand they make a bet that is half of the big blind. They are the second to last to talk in the Pre-Flop phase, just before the big blind.
- **Big Blind (BB):** Second player to talk after the Pre-Flop phase; furthermore at the start of the hand they make a bet that is equal to the big blind. They are the last player to talk in the Pre-Flop phase.
- **Under the Gun (UTG):** First player to talk in the Pre-Flop phase, and the third assuming that the two blinds do not fold. It is the position with the tightest opening ranges, because there is no information from the other players.
- **Mid Position (MP):** Intermediate positions (for boards with more than six players, there are several players in this position) after UTG. Here the range will be a little wider.
- **Cut Off (CO):** Player after MP with wider range because there is information about the previous players.
- **Button or Dealer (BTN):** Player that talks before the blinds. The range in this position will be wider.

2.3.3. Actions

- **Fold:** Action of not playing a hand and “folding” the cards, not taking the bets from a previous player. Only done when the cards aren’t favourable to win.
- **Call:** Action of taking the bet and equalling it to the bet from a previous player to continue to the next phase. If it is the River, see who has the best hand.
- **Limp:** A special action of calling a bet equal to the big blind without raising.
- **Raise:** Action of raising a previous bet.

2.3.4. Ranges

All ranges vary depending on the position of the player.

- **Opening Range (OR):** Range of hands on which we will raise the bet when the hand comes to us without being raised, or just with the blinds and the limpers.

- **3-Bet range (3BET)**: Range of hands on which we raise a bet from a previous player.
- **Fold to 3-Bet range(F3BET)**: Range of hands on which we fold when another player makes a 3-Bet to our bet.
- **4-Bet range(4BET)**: Range of hands on which we raise a bet from a 3-Bet from a previous player.
- **Fold to 4-Bet range(F4BET)**: Range of hands on which we fold when another players make a 4-Bet to our bet.
- **Cold Calling (CC)**: Range of hands to call a previous bet.

2.3.5. Other concepts

Values that will be used from the PokerTracker 4 database

- **VPIP**: Percentage of cases for which the player puts money in the pot without obligation before the “Flop”.
- **“Pre-Flop Raise” (PFR)**: Frequency of a player raising the bet in the Pre-Flop.

Some of the previous concepts of ranges also appear as data in the Pokertracker database and will reference the percentage for a player to do that action.

These two concepts are relevant to see the ranges depending of the rival we face and the position on the table:

- **“In Position” (IP)**: To be in position facing another player means that we talk after that player, for example the player in BTN is IP facing UTG.
- **“Out Position” (OP)**: To be out of position facing another player means that we talk before that player, for example the player in UTG is OP facing BTN.

2.4. Game Modality

As we said before, there are many variants of poker. In this Graduate Thesis we center on the variant of NLHE in the “Pre-Flop” phase and the tools needed. In this phase each player, must decide to bet or not in function of the cards in their hand and the probabilities to make a hand.

Furthermore we have chosen the scenario of a table with nine active players due to recreational players usually playing big tournament tables that have that number of players.

2.5. Frame of work

All algorithms and functions in this Graduate Thesis have been developed in Java. The frame of work aims to help recreational players that play online poker. For this, users have a live action recommender and a customizable trainer to practice different situations and improve their game.

The recommender can also be customized. The ranges can be modified in order to, once the player know how to play and their personal ranges, they can modify the existing ranges or creating new ranges.

Finally, we have used external libraries such as Tess4J (Nguyen, 2016) to recognise the number of the bets of the players on the table from an image. There are also some restrictions due to the difficulty and the time it took to adapt to the changes in the PokerStars user interface (PokerStars.es, 2016b). These restrictions are:

- The player must select in the PokerStars options to be seated on the bottom center part of the table.
- The player must select from the user options to use the deck theme with four colours and white numbers.
- No other window can cover the corners of the Pokerstar application (image 2.2) since they are needed to locate the window.



Figura 2.2: PokerStars window

Capítulo 3

Proceso de desarrollo

*Si la suerte no influyera en el poker, yo
siempre ganaría.*

Phil Hellmuth

RESUMEN: En este capítulo se explica cómo hemos trabajado de forma conjunta los integrantes del trabajo de fin de grado.

3.1. Introducción

Teóricamente todo software desarrollado debería conllevar una documentación que se iría realizando a la vez que el proceso de desarrollo. Desgraciadamente en la mayoría de las ocasiones esto no sucede así; lo más común es generar al final un documento recogiendo la funcionalidad del código. Para el caso en el que el proyecto está integrado por más de una persona, la situación se complica de cara a asegurar que todo siga adelante de forma clara y robusta, ya que implica estar en constante comunicación con los demás integrantes del equipo. Para solucionar este problema se siguen procesos de desarrollo con una definición de los pasos y objetivos para cada fase hasta llegar al final.

Por lo anterior, al ser dos personas trabajando en este proyecto, decidimos usar una metodología ágil basada en Scrum para asegurar una comunicación efectiva y eficiente. La metodología Scrum tiene muchos beneficios, como su gran flexibilidad a cambios, mayor calidad de software, mayor productividad, o disponer de una predicción de tiempos para medir la velocidad a la que se desarrolla.

La metodología Scrum se basa periodos de un tiempo determinado llamados “Sprints”, con reuniones periódicas y objetivos a terminar. Los Sprints pueden ser de tiempo variable, dependiendo de los objetivos del Sprint. Esta metodología es una de las más usadas en las empresas del sector de desarrollo de Software, ya que permite una integración rápida y con objetivos finales de cara a un cliente que quiere obtener su producto y, a partir de ahí, mejorarlo en cada Sprint.

Desde la primera reunión decidimos dividir el trabajo de tal forma que ambos pudiéramos avanzar sin tener que preocuparnos de la parte del compañero, para así agilizar el proceso y no ralentizarlo con conversaciones intermedias. Para almacenar los avances en el código usamos la plataforma Github como sistema de control de versiones.

3.2. Metodología Scrum

En la versión pura de esta metodología el ciclo que se sigue en cada Sprint tiene las siguientes reuniones:

- **Daily Scrum:** reunión diaria para ver cómo va el proyecto.
- **Reunión de planificación del Sprint (Sprint planning meeting):** reunión al inicio de cada Sprint para seleccionar el trabajo que se hará.
- **Reunión de revisión del Sprint (Sprint Review Meeting):** reunión en la que se revisa el trabajo realizado.
- **Retrospectiva del Sprint:** reunión en la cual se opina sobre el Sprint para mejorar.

El Sprint es el periodo en el cual se lleva a cabo el trabajo en sí. Suele tener una duración constante definida por el equipo en base a la propia experiencia, pero en cada Sprint se puede modificar en función del ritmo del equipo o de los objetivos por fechas.

3.3. Aplicación de Scrum al trabajo de fin de grado

Al ser un grupo de dos personas, nos hemos basado en la metodología Scrum pero adaptándola a nuestra situación. Hemos eliminado las reuniones diarias ya que, al ser solo dos, en cualquier momento teníamos disponibilidad para hablar el uno con el otro y tratar los problemas según surgieran. Esto mismo es también aplicable a la retrospectiva.

Por tanto, nos quedamos con los beneficios de las reuniones de planificación y las de revisión. Al inicio decidimos empezar con un Sprint inicial de 60 días debido a la necesidad de investigar las herramientas y tomar decisiones, para después continuar con Sprints de 30 días.

Los Sprints realizados fueron los siguientes:

1. Sprint Inicial [25/11/2016-25/1/2016]

- En la *reunión de planificación* se decidió que ambos miembros del equipo nos dedicaríamos a la búsqueda de información sobre las salas de póker online e investigación sobre estrategias y las primeras aproximaciones en reuniones para darle forma al recomendador.
- En la *reunión de revisión* se decidió usar la sala PokerStars por ser, con mucha diferencia la más popular, tener una interfaz adaptable y, en un primer momento, la facilidad para sacar los datos que necesitábamos.

2. Sprint de Inicio de desarrollo [25/1/2016-25/2/2016]

- En la *reunión de planificación* se decidió separar el trabajo en dos partes debido a unos cambios en la interfaz de PokerStars que hicieron que no se pudieran obtener los datos de manera sencilla. El reparto fue el siguiente, y se mantendría hasta el final del trabajo: el alumno Juan Carlos Marco González se encargaría de hacer la implementación en Java del recomendador y el alumno Iván Martín Herrero se encargaría de hacer un reconocedor de imágenes para conseguir la información en el momento del turno del jugador.

Partiendo de aquí, ambas funcionalidades se reestructuraron y subdividieron en varias tareas para los siguientes sprints. Para el Sprint actual se decidió que Juan Carlos Marco implementaría las tablas de Janda (Janda, 2013) pasadas al código y las funciones del recomendador para procesar la información llegada del Reconocedor de imágenes. Por su parte, Iván Martín implementaría el reconocedor de pantalla devolviendo la información de las cartas en mano, ciegas y posición.

- En la *reunión de revisión* cada uno presentó su parte pero se tenían ciertas dudas. En la parte del recomendador había algunas jugadas que no estaban claras y se decidió preguntar al tutor para saber cómo había que leerlas, de cara a saber cómo implementarlas. En el apartado del reconocedor se llegó con una solución para un tamaño de pantalla fijo y sin obtener aún las ciegas, por problemas al interpretar los números.

3. Sprint de mejoras [25/2/2016- 25/3/2016]

- En la *reunión de planificación* se decidió dedicar un sprint a mejorar lo que ya se tenía de las jugadas, con los cambios sugeridos por el tutor, y en la parte del reconocedor de imagen hacer que capturara los datos independientemente del tamaño de la pantalla.
- En la *reunión de revisión* se mostró una primera versión de cómo funcionaba el recomendador en directo con las mejoras en las jugadas, pero aún con la versión anterior del reconocedor ya que al extender la funcionalidad para cualquier tamaño, llevó tiempo de investigación volver a hacer los algoritmos para encontrar la pantalla de juego, las cartas, las posiciones y las ciegas, trabajo que se continuaría en el siguiente Sprint.

4. Sprint de Reconocedor y entrenador [25/3/2016- 25/4/2016]

- En la *reunión de planificación* se decidió, con la ayuda del tutor, hacer un entrenador de jugadas como opción para aprender cómo jugar al póker y reaccionar ante las situaciones más comunes o aleatorias. En el apartado del reconocedor de imagen se decidió finalizarlo con la investigación hecha para los distintos tamaños y con la librería Tess4J para leer los números de las ciegas.
- En la *reunión de revisión* se llegó con todo realizado y funcionando, tanto el entrenador como el reconocedor para el recomendador.

5. Sprint de mejoras y memoria [25/4/2016 - 25/5/2016]

- En la *reunión de planificación* se decidió seguir mejorando y haciendo debug del software para casos distintos, ya ambos miembros del equipo trabajando en lo mismo, buscando los errores que pudieran surgir, además de investigando para mejorar el reconocedor y optimizarlo frente a jugadores explotables. Así mismo, se decidió comenzar la escritura de la memoria y darle forma. El coordinador de la redacción fue el alumno Iván Martín.

6. Sprint de Finalización [25/5/2016-Final]

- Se terminó de añadir el recomendador adaptativo con las llamadas a la base de datos de PokerTracker. Además, se completó la escritura de la memoria con los últimos cambios pertinentes y se procedió a su entrega al tutor.

3.4. Contribuciones de Juan Carlos Marco González

3.4.1. Servidor PokerStove

Se empezó a crear una aplicación de servidor que ofrecería, mediante la invocación de métodos en remoto de Java, Java RMI, la posibilidad de calcular el *equity* desde la aplicación cliente, utilizando una herramienta ya existente llamada PokerStove, capaz de calcular millones de manos en apenas segundos. Sin embargo, nos encontramos con el hándicap de no saber cómo usarla por medio de la terminal. Tras un periodo de investigación para determinar si existía alguna forma de invocar al programa desde la terminal, y no encontramos nada al respecto. Se tomó la decisión de usar éste en su modo normal, es decir, el gráfico.

Para ello, se comenzó a investigar cómo se podían usar aplicaciones con entorno gráfico desde Java y, al fin, se dio con la respuesta. Ésta se corresponde con la clase Robot de Java, la cuál permite usar los eventos normales de uso para comunicarse con la maquina, como son: mover el ratón, hacer click, escribir, etc.

En la parte del servidor se decidió crear una interfaz con métodos acordes a las prestaciones que da la aplicación PokerStove (cómo introducir los rangos de los jugadores en sus respectivas casillas, cambiar entre la generación de todas las manos o modo Montecarlo, etc.). De esta forma es posible que desde la aplicación cliente, se pudiera usar PokerStove, en remoto, con la garantía de recibir los datos de forma rápida y poder usarlos en nuestro recomendador de póker.

A continuación se presentó un nuevo hándicap: ¿cómo conectar el servidor con la aplicación cliente? Para esta cuestión se comenzó a investigar la conexión remota en Java, y se llegó a dos posibles soluciones: usar Java RMI, el cuál permite manejar objetos instanciados en la memoria de otra máquina, de forma que desde la aplicación cliente se puede utilizar como si estuviera en la propia máquina; o, como segunda opción, usar Sockets, los cuales por medio de la red, envían mensajes con comandos y un servidor que al recibir estos mensajes, los clasifica y responde acorde a los comandos creados.

Finalmente, se eligió la primera opción, Java RMI, dado que ya se tenía hecha una interfaz la cuál generaba el comando concreto para PokerStove, y por tanto adaptar ésto a Java RMI era una cosa trivial y por consiguiente se ganaría bastante tiempo.

El funcionamiento de Java RMI adaptado a nuestro código es el siguiente: se basa en dejar corriendo el programa servidor **PokerStoveRMI** en la máquina donde está la aplicación de PokerStove y ejecutar el **clienteRMI** desde otra máquina. Como en el proyecto RMI de la parte cliente debe aparecer tanto el código del cliente, como la parte del servidor (para poder usar objetos instanciados en otra máquina) en el cliente se llama a la clase **servidorRMI** del proyecto, pero realmente no se está usando ésta, sino que se usa una interfaz, la cuál abstrae al usuario de la realidad, y porque de hecho se está ejecutando el código de otra máquina como si estuviera en la tuya propia. Entonces al llamar a los métodos creados para PokerStove, éste se ejecuta en remoto y se puede ver cómo el entorno remoto es manejado y devuelve la información oportuna.

3.4.2. Maestro

El maestro es parte crucial del proyecto. Podemos distinguir dos tipos de funcionalidades del maestro:

- Decidir en qué tipo de jugada te encuentras y decirle al jugador qué acción debe tomar, FOLD, CALL o RAISE, apoyándose en la lógica implementada sobre detección de jugadas y los rangos de cartas introducidos para cada tipo de jugada.
- Explotar las debilidades de los rivales analizando sus jugadas almacenadas en PokerTracker 4 de forma que se pueda extraer sus tendencias.

Para la primera parte, se han usado los conocimientos almacenados en el libro “Applications of No-Limit Hold’Em” (Janda, 2013) de Matthew Janda sobre rangos en situaciones, empezando por crear una herramienta gráfica que almacena los rangos de Matthew Janda dependiendo de la situación. Gracias a la asignatura impartida por nuestro tutor Manuel Núñez, “Herramientas informáticas para juegos de azar”, que cursamos el año pasado, ya disponíamos de una herramienta capaz de almacenar rangos, aunque se tuvo que adaptar a la nueva funcionalidad, puesto que ahora no solo se almacenaban rangos, sino que eran rangos para diferentes situaciones.

Gracias a esta herramienta y sus correspondientes modificaciones, se logró crear una herramienta gráfica capaz de satisfacer nuestras necesidades, y capaz también de reducir el tiempo para volcar la información del libro de Janda a nuestro Proyecto.

Se crearon dos clases para esta nueva funcionalidad:

- **VentanaRangoPersonalizado.** Contiene una cuadrícula con todos los rangos de cartas que hay en la baraja francesa para No-Limit Hold'Em, las situaciones que se pueden generar en una mesa y los rangos ya almacenados previamente. Esta ventana da la posibilidad de almacenar de forma gráfica el rango concreto para la situación concreta e incluso modificar algún rango ya establecido (opción no recomendable puesto que ya están introducidos los rangos que usa Janda)
- **VentanaRangoEscrito.** Da la posibilidad de introducir por texto el rango que el usuario quiera y generarlo en la cuadrícula de rangos de *VentanaRangoPersonalizado*, opción mas rápida que la anterior puesto que permite introducir mas rangos en menor tiempo, aunque también requiere mayores conocimientos sobre póker para la introducción de éstos por texto.

Ambas clases tienen un aspecto similar a PokerStove y, por consiguiente, un usuario que sepa usar PokerStove sabrá usar también esta herramienta.

También existe un archivo en la raíz del proyecto llamado *Configuracion-Rangos*, el cuál esta configurado de la siguiente forma:

`Situación_de_mesa_de_póker:rango_a_usar`

Este archivo permite asignar diferentes rangos a los preestablecidos en las diferentes situaciones, ya que aunque como dijimos anteriormente no es una opción recomendable, es cierto que algún jugador experimentado puede creer oportuno cambiar algún rango en alguna situación específica.

Es importante decir que todos los rangos generados se almacenan en la carpeta **rangosPersonalizados**, así que para configurarlos tan solo habría que introducir el nombre del rango que se quiere usar en la situación que se quiera usar.

Después de crear la herramienta para la introducción de los rangos, se tuvo que introducir la lógica necesaria para detectar en qué situación de la mesa estábamos. Para ello, se creó en el maestro un método que se llama **eleccionPreFlop**, que a su vez tiene definidas todas las situaciones que se pueden dar en una mesa de póker dependiendo de la posición ocupada en ella. Al estar definidas todas las situaciones dentro de la lógica del método, tan solo hay que comprobar cuantas subidas ha habido en la mano, la última posición desde donde se subió y la posición que ocupamos en la mesa (se crearon métodos que indican estos datos dentro del propio maestro), y este método devuelve la acción óptima a realizar.

Por otro lado, para la parte de explotación de los contrincantes se investigó sobre el programa PokerTracker 4. PokerTracker 4 es un software que

analiza las manos generadas por las distintas salas de póker del mercado y almacena en su BBDD las acciones que realizan los usuarios en las partidas que juegan, creando así tendencias de éstos en diferentes situaciones del juego, las cuales vamos a intentar explotar en el juego Pre-Flop.

Lo primero que se buscó fue saber cómo está estructurada la BBDD de PokerTracker 4, para entonces empezar a indagar sobre cómo realizar consultas a ella. Se empezó a investigar sobre métodos para que, dado un script SQL, se generara de forma gráfica la estructura de la BBDD para así saber por donde empezar a investigar. Se encontraron varias herramientas que realizaban esta labor, pero no servían para todo tipo de BBDD, y ésta en concreto está hecha en PostGreSQL. Al final se dio con la solución: SchemaSpy, una herramienta capaz de generar la estructura de la BBDD en formato html, la cual está almacenada en la ruta *documentación/Estructura BBDD PokerTracker*.

Gracias a la generación de esta estructura, se pudo investigar cómo podrían realizarse las consultas oportunas a la BBDD. Haciendo pruebas e investigando por la red, se llegó a la conclusión de que unas tablas clave para la consulta de estadísticas son *tourney_cache* y *cash_cache*, aunque nosotros solo utilizamos la primera, puesto que en principio nuestro trabajo se concentra en ayudar a los jugadores de torneos.

Se aislaron los siguiente valores de cada jugador:

- VPIP,
- P_R,
- P_3BET,
- F_3bet_GENERAL,
- F_P_3BET,
- F_F_3BET,
- F_T_3BET, y
- F_R_3BET

Estos valores se obtuvieron gracias a la siguiente consulta:

```
SELECT 100*SUM(cnt_vpip) /
(SUM(cnt_hands) - SUM(cnt_walks)) as VPIP,
100*SUM(cnt_pfr) / SUM(cnt_pfr_opp) as P_R,
100*SUM(cnt_p_3bet) / SUM(cnt_p_3bet_opp) as P_3BET,
```

```

ROUND(100*(((SUM(cnt_p_3bet_def_action_fold) +
SUM(cnt_f_3bet_def_action_fold) +
SUM(cnt_t_3bet_def_action_fold) +
SUM(cnt_r_3bet_def_action_fold))):float /
((SUM(cnt_p_3bet_def_opp) +SUM(cnt_f_3bet_def_opp) +
SUM(cnt_t_3bet_def_opp) + SUM(cnt_r_3bet_def_opp))))
as F_3bet_GENERAL,
ROUND(100*(SUM(cnt_p_3bet_def_action_fold)::float / CASE
WHEN (SUM(cnt_p_3bet_def_opp) = 0 ) THEN 1
else SUM(cnt_p_3bet_def_opp) END)) as F_P_3BET,
ROUND(100*(SUM(cnt_f_3bet_def_action_fold)::float / CASE
WHEN (SUM(cnt_f_3bet_def_opp) = 0 ) THEN 1
else SUM(cnt_f_3bet_def_opp) END)) as F_F_3BET,
ROUND(100*(SUM(cnt_t_3bet_def_action_fold)::float / CASE
WHEN (SUM(cnt_t_3bet_def_opp) = 0 ) THEN 1
else SUM(cnt_t_3bet_def_opp) END)) as F_T_3BET,
ROUND(100*(SUM(cnt_r_3bet_def_action_fold)::float / CASE
WHEN (SUM(cnt_r_3bet_def_opp) = 0 ) THEN 1
else SUM(cnt_r_3bet_def_opp) END)) as F_R_3BET
FROM tourney_cache
WHERE id_player =(SELECT id_player FROM player
WHERE player_name='&#39;&quot;+ jugador + &quot;&#39;)&quot;);

```

Para la consulta de datos, se ha tenido que modificar el fichero de configuración de la BBDD de PokerTracker 4 para que permitiera la conexión remota a esta. También se ha creado una clase **IOReceiveFile** para el servidor donde esté PokerTracker 4 instalado, que se encarga de recibir el fichero de manos que se genere en el cliente y almacenarlo en el directorio de almacenamiento de manos de PokerStars; PokerTracker 4, al detectar una nueva mano, la procesa y la almacena en su BBDD, así teniendo siempre la BBDD actualizada para poder realizar las consultas con los datos más recientes posibles.

En el cliente se han creado también las clases **IOSendFile**, que se encarga de enviar el fichero de manos al servidor, y **FilePokerStars** que se encarga de tratar los datos que se generan en una mano de PokerStars (capáz de leer los nombres de los jugadores que están jugando con nosotros, leer stacks de éstos, posiciones, etc., entre otras opciones) para así posteriormente poder crear las consultas contra PokerTracker 4 con los jugadores pertinentes y así poder explotar sus debilidades.

3.4.3. Trainer

El “trainer” es un instrumento de aprendizaje para jugadores noveles que están empezando en el mundo del póker. Este entrenador proporciona la posibilidad de recrear cualquier jugada que se pueda dar en una mano y, así, poder afinar nuestro rango dependiendo de la situación y la posición elegida en la mesa.

Los modos de entrenamiento que alberga este trainer son los siguientes:

- **Normal:**

Este modo sería el que permite modificar cualquier aspecto de la mesa, es decir, permite cambiar las acciones que hacen los oponentes antes de tomar la propia decisión, elegir el sitio que se desea tener en la mesa en esa mano y el rango de cartas que se quiere usar.

Una vez elegido esto, solo quedaría elegir la acción a realizar y pulsar el botón evaluar. Al pulsarlo, se llama al maestro, y éste detecta qué situación hay en la mesa. Al detectar esta situación, se lee el rango correspondiente a ésta (normalmente serán los de Mathew Janda, aunque como dijimos anteriormente, siempre se pueden modificar) y el maestro valora si el rango y acción elegidos corresponden con la opción óptima en esta situación, para finalmente lanzar su veredicto y confirmar si fue una acción correcta o incorrecta.

- **Aleatorio:**

Este modo propone generar una jugada aleatoria con un rango aleatorio, por lo cual se pueden generar situaciones que se den muy pocas veces en el póker, las cuales pueden servir para el perfeccionamiento de la forma de jugar. Por el contrario, es un modo demasiado avanzado para un jugador recreacional por lo que, para suplir esta *deficiencia*, se han creado los siguientes dos modos, que son las situaciones que más se dan en el póker.

- **Open Raise:**

Este modo genera situaciones aleatorias de Open Raise, es decir, genera situaciones en las que ningún contrario antes que el usuario haya realizado ninguna subida, y da la opción de entrenar los rangos de apertura desde una posición aleatoria.

- **3-Bet:**

Este modo genera situaciones aleatorias de 3-Bet, las cuales consisten en elegir un contrario al azar delante del usuario que haga raise, y finalmente elegir una posición propia y generar un rango aleatorio para dar la posibilidad de entrenar su rango 3-Bet de diferentes situaciones.

3.5. Contribuciones de Iván Martín Herrero

3.5.1. Idea inicial del proyecto

El objetivo de este proyecto, así como los contenidos que tendría, surgen de la idea de las clases de años anteriores de la asignatura impartida por Manuel Núñez, también tutor del proyecto, “Herramientas Informáticas para los juegos de azar”. Al finalizar la asignatura, y tras todo el año estudiando las herramientas y el juego para usarlas correctamente, surgió la idea de hacer un “ayudante” para los jugadores que no sepan o se estén iniciando en el juego del póker online y que no sepa usar estas herramientas de manera óptima, al menos durante el tiempo de aprendizaje, para que puedan ir más rápido.

Al parecerle correcta a Manuel la idea para el trabajo de fin de grado, nos puso en contacto a los dos integrantes del equipo, ambos interesados en el póker y en el objetivo del trabajo.

3.5.2. Gestión del proyecto

Gracias al trabajo en la empresa del alumno Iván Martín Herrero, hubo una familiarización con distintos modelos de gestión de proyectos, sobre todo aquellas metodologías ágiles usadas ampliamente hoy en día en las empresas como pueden ser el desarrollo de software adaptativo, la programación extrema, o Scrum. Al ser un equipo pequeño, y dada la escasez de tiempo, nos decantamos por utilizar Scrum, ya que permite definir unos objetivos por reunión y concretar con tiempo la siguiente.

Por tanto, empezó la gestión para llevar preparado en cada reunión todo lo necesario para hablar de los objetivos para el siguiente Sprint, las ideas generales de cómo desarrollarían estos objetivos y una estimación temporal. En cada reunión se estudiaba lo realizado, y de lo que hubiera faltado por completar en ese Sprint, más las ideas traídas desde casa para continuar, se decidían los objetivos del siguiente Sprint.

En esta gestión también se deben incluir algunas tomas de decisiones respecto a ciertas partes del proyecto, cómo realizarlas y dedicar tiempo a preparar las reuniones.

3.5.3. Realización y maqueta de la memoria

Al haber gestionado el proyecto, Iván Martín tenía una mayor información de cara a poder realizar la memoria, por ejemplo porque al haber preparado las reuniones, tenía los objetivos según iba avanzando; incluyendo

también que al estar familiarizado con hacer documentación de los proyectos para la empresa, tenía mayor conocimiento para hacerlo de forma más eficiente.

Para la maquetación se usó \TeX IS, una plantilla de \LaTeX (Oetiker et al., 1996), un sistema de composición de textos orientado a la creación de documentos escritos con alta calidad tipográfica. \TeX IS está desarrollada por dos profesores de la Facultad: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín (Gómez-Martín y Gómez-Martín, 2009). Esta plantilla facilitó la maquetación ya que aporta un sistema común a los demás documentos de la facultad, ayuda a separar las secciones y aporta un estilo distintivo a la portada y al proyecto en general.

3.5.4. Reconocedor de imagen

Tablero de Juego

Llamaremos tablero de juego a la ventana de PokerStars con una partida en curso (figura 3.1) y dentro de la cuál vamos a leer las distintas opciones que necesitaremos de cara a decidir qué hacer en cada jugada.



Figura 3.1: Tablero de juego de PokerStars

Para encontrar la posición de esta ventana se decidió usar el logotipo de

PokerStars ya que nos daba la información de dónde empezaba la ventana y, como en un primer momento se decidió hacerlo con la pantalla de tamaño predeterminado, nos daba toda la información necesaria para encontrar el tablero de juego. Se procedió a desarrollar este algoritmo para encontrar la pantalla mediante la extracción de la imagen a buscar y un algoritmo de búsqueda para encontrar un parecido con esta imagen: se buscaba que coincidiera la imagen al completo, mediante una comparación del color de los píxeles de la imagen tomada mediante una captura de pantalla y la imagen extraída. Al tratarse de una imagen pequeña, la comparación no llevaba mucho tiempo, pero si el recorrido que se realizaba para encontrar el inicio.

Posteriormente, se mejoró el código de tal forma que sólo comparaba la imagen al completo si, una vez encontrado el primer píxel coincidente, el último de la imagen extraída coincidía con el de la captura, avanzando el tamaño de la imagen extraída; en caso contrario, se seguía con la búsqueda habiendo hecho solo dos comparaciones. Una mejora posterior consistió en tener en cuenta que determinada la coincidencia de ambos extremos, sólo se compararían con las respectivas posiciones en la captura de pantalla: el píxel central, el superior, el inferior, el izquierdo y el derecho respecto al central (es decir, la cruz formada por los 5 píxeles mencionados). Esto resultaba más eficiente y todavía efectivo, ya que la probabilidad de encontrar ese patrón en otro lugar de la pantalla era muy baja, al ser un logotipo.

Una vez realizada la última versión, al realizar la unión con la lógica que leería los datos, aunque solo fuera el encontrar el tablero de momento y una primera versión de la lógica, se vio que en una versión anterior de Windows la ventana era diferente y, aunque aparecía el logotipo de PokerStars, aparecía descentrado y además las dimensiones no eran las mismas debido a la resolución de pantalla. Por tanto en la siguiente reunión del Sprint se decidió cambiar el método, lo que conllevó una investigación adicional para encontrar de otras formas de hacerlo, y además, hacerlo escalable.

Tras un periodo de investigación y de ensayos y errores, ya que la pantalla al cambiar de tamaño no seguía ninguna proporción, se llegó a la versión funcional y escalable que reconocería la posición y el tamaño del tablero. Para ello se descubrió que el chat y el tamaño del campo de escritura del mismo situado en la parte inferior izquierda de la pantalla siempre mantenían el mismo tamaño independientemente del tamaño exterior de la pantalla, por lo que se tomó como punto de partida.

Ahora faltaba encontrar el tamaño tanto de ancho como de largo de la pantalla. Esta parte fue más tediosa ya que, dependiendo del tamaño de la ventana, los colores y formas de las esquinas superior e inferior derechas y la

superior izquierda cambiaban demasiado como para tomar un patrón claro o una imagen de partida como en el caso inicial del proyecto.

Finalmente, tras probar varios métodos, se encontró un patrón en la esquina superior izquierda que permitió establecer el alto del tablero de juego así como el punto inicial desde el que se procedería a cortar la imagen para luego tratar el tablero de forma autónoma.

Nos quedaba entonces encontrar el ancho pero, una vez tenemos la altura en la que se empieza, solo había que seguir esa línea de píxeles, ya que en algún punto se terminaría para dar paso a la pantalla de escritorio del usuario o marco de la ventana. Para ello se encontró otro patrón en esa esquina, que cuando cambiaba después de una línea en un tono de gris a otro color, se sabía que era el final, y por tanto ya teníamos el ancho del tablero.

Una vez teníamos el punto inicial, alto y ancho de nuestro tablero, pudimos proceder a hacer un recorte de la captura inicial de la pantalla para solo tener el tablero de juego y leer los demás datos que necesitaremos, ya que aunque la proporción no es siempre la misma con el tamaño (esto lo volveremos a ver dentro de cada uno de los reconocimientos de elementos siguientes) difiere poco una vez tomado los valores correctos del tablero de juego.

Ficha de posición

Llamaremos ficha de posición al círculo con el as de picas rojo que esté en el tablero e indica quien es el Dealer (repartidor) y a partir del cual se ponen las ciegas.

Para encontrar esta ficha se sacó una imagen de ella para buscar dentro del tablero (figura 3.2). Al principio se sacaron todas las posibles posiciones (SB, BB, UTG, EP, MP, CO, BTN) con las coordenadas donde tendría que estar la ficha de posición en cada una, ya que el tamaño de la ventana era fijo y se iba probando si coincidía por posición hasta que una encajara.



Figura 3.2: Ficha de posición en PokerStars

Tras ver los anteriormente mencionados casos en los que la pantalla podría tener tamaños distintos, este método resultó poco fiable, por lo que se procedió a investigar nuevas formas de buscar la ficha de posición, descubriendo que el tamaño de la ficha también variaba con el tamaño de la

pantalla, por lo que se decidió coger una imagen de cada uno de los tamaños posibles, seis en total.

Acto seguido, dado que la posición dentro de la pantalla también variaba según el tamaño, y teniendo en cuenta que el tamaño era prácticamente proporcional y el error despreciable, se decidió que se acotarían las zonas en función del tamaño de la ventana usando porcentajes para delimitar las búsquedas de las posiciones. Una vez delimitada, en la porción objetivo se hace una búsqueda de la imagen de la ficha de posición, empezando por la de tamaño más común y así sucesivamente por posición hasta que la ficha de tamaño correcto es encontrada, devolviendo la posición en la que nos encontramos.

Además, este método se aprovecha para saber en qué tamaño de tablero nos encontramos ya que las cartas de la mano cambian de tamaño a la par que esta ficha. No se puede usar el tamaño que hemos sacado para el tablero porque los cambios entre una ficha y la de siguiente tamaño apenas eran apreciables cuando se investigó.

Cartas de la mano

Llamaremos cartas de la mano a las dos cartas que se muestran al jugador durante la mano y que serán exclusivas de dicho jugador para completar las jugadas.

Para encontrar estas cartas, como al principio se supuso un tamaño fijo de la ventana, se buscaban las coordenadas dentro del tablero de juego. Para obtener el palo de la carta se optó por coger el color de un píxel de dentro de la carta y en función del color asignar el palo ya que al ser constante el punto del que lo sacamos para cada uno de los posibles palos (picas, diamantes, corazones, tréboles) sería siempre el mismo color. Para el número abstraeríamos la zona superior izquierda de la carta, que es donde se encuentra, y se cuenta el número de píxeles en blanco que hay para el número.

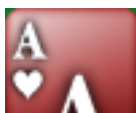


Figura 3.3: As de Corazones

Tras el cambio a tablero de juego escalable, volvimos a sufrir problemas similares a los casos anteriores, ya que el tamaño de las cartas y sus coordenadas dentro del tablero cambiaban en función del tamaño del tablero.

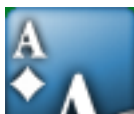


Figura 3.4: As de Diamantes

Tras comprobar que el cambio de tamaño era análogo al de la ficha de posición, se crearon unas constantes asociadas con las posiciones de las cartas para cada tamaño, de forma que, en función del tamaño obtenido en el método de la búsqueda de la ficha, pudiéramos usar las coordenadas adecuadas.

Para obtener el palo se continuó usando el mismo método, dado que al ser las coordenadas fijas para una carta del mismo palo nos devolvería en ese pixel el mismo color.

En cuanto al número, se siguió cortando la imagen de la carta para sacar solo la zona del número en función del tamaño, y se empezó a usar la librería Tess4J, una librería de reconocimiento de texto y números en imágenes. Al principio fallaba, ya que muchas veces no conseguía leer bien el número, ya que al hacerlo por reconocimiento de patrones y comparación con su diccionario, a veces eran muy parecidas y daba resultados erróneos. Para resolver este problema se hicieron casos excepcionales dentro del método para sacar el número y calibrar estos errores.

Tras lo anterior hubo una mejora que permitió hacer más rápido el reconocimiento; los tamaños de las cartas se dejaron de guardar en función del tablero, y pasaron a almacenarse como porcentaje en función del tamaño del tablero obtenido en el paso de recortar el tablero de juego. Por ello hubo que cambiar el modo de obtener el palo, ya que no siempre se obtenía ahora el mismo pixel en la posición de la carta, aunque se siguió el mismo proceso estableciendo rangos del color (la función que nos daba el color del pixel los daba en tipo entero) y así diferenciarlos. La mejora en el número consistió en tener en cuenta que los rangos usados para obtener el palo también se usaron para convertir esas zonas a pixeles en negro y dejó solo en color claro el número de la carta para que Tess4J pudiera sacarlo de forma más eficaz.

Apuestas de rivales

Llamaremos apuestas de los rivales a los montones de fichas que aparecen delante de las personas a los que nos enfrentemos. Aunque desarrollamos un método que devuelve las de todos, en la lógica solo se tienen en cuenta hasta nuestra posición. Es decir, si estamos en MP solo se van a tener en cuenta las de SB, BB, y EP que son las personas que han jugado ya, las de los demás

será cero.

Se hizo un primer intento con la librería Tess4J para la versión de tamaño fijo, pero fallaba bastante al distinguir los números; se necesitaba entrenar más el conjunto de diccionarios o hacer reglas excluyentes. Para obtener estos números se recortaba la porción del tablero donde estaban las ciegas y se pasaba el archivo a Tess4J.

Pero antes llegó el hacerlo escalable, y tras los avances realizados para obtener los números de las cartas, se aplicó el mismo algoritmo consistente en convertir en píxeles negros todos aquellos que no formaran parte del conjunto de colores en el que se mostraban los números. Además, se siguió la misma forma de capturar las zonas, investigación de las posiciones y establecer a partir de que porcentaje se encontraban para buscarlo independientemente del tamaño del tablero.

Ciegas, apuestas y bote

Por último nos queda sacar las apuestas hechas en la mesa hasta nuestra posición. Para ello se investigó un procedimiento que consistía en capturar la posición, de las apuestas de cada jugador para aplicar a continuación el mismo método que con las cartas para obtener el valor de las apuestas.

Al principio, como en los casos anteriores, estaba establecido dónde se encontraban pero, tras el cambio para permitir que los tableros fueran escalables, se pasó a porcentaje en función del tamaño total del tablero de juego. Para cada uno de los jugadores se toma una imagen de donde debería estar su apuesta, y ésta se pasa a un método para encontrar los valores, si no hay apuesta devuelve 0, y si la hay entonces devuelve la apuesta encontrada.



Figura 3.5: Apuesta de un jugador

En este apartado se aplicó la segunda versión de la optimización consistente en volver los píxeles negros, a excepción de los que fueran los correspondientes a los números para que así la librería Tess4J los reconociera mucho mejor.

Por último, el bote total de la mesa se saca de manera análoga a las apuestas (figura 3.6).



Figura 3.6: Bote total de la mesa

Capítulo 4

Diseño, uso y resultados

*El juego del poker no consiste en ganar
dinero, sino en tomar buenas decisiones.
Si tomas la decisión correcta, ganarás
dinero.*

Annie Duke

4.1. Diseño del software

El diseño de este software se planteó como un programa amigable y fácil de usar por los usuarios de manera intuitiva. Para ello se hizo el siguiente diseño.

Un menú principal con las opciones del software de tal forma que pulsando en cada botón el usuario accede a la herramienta que se quiera usar en ese momento. De cualquier ventana se vuelve al menú principal cerrándola, lo que se representa en los diagramas como una X . En el primer diagrama de flujo (figura 4.1) se muestran los distintos caminos que se pueden seguir desde el menú principal.

Cada uno de los flujos se explicará con más detalle en las siguientes secciones.

4.1.1. Flujo del Recomendador

Tras llegar a la ventana del recomendador se comprueba si el usuario ha añadido previamente una ruta a sus manos de Pokerstars; en caso contrario le pedimos que lo haga ahora.

Una vez tengamos la ruta de las manos, el usuario podrá poner en marcha el recomendador con el botón de Start, y el software empezará a buscar la mesa en la pantalla y, mientras no la encuentre, seguirá esperándola. Una vez

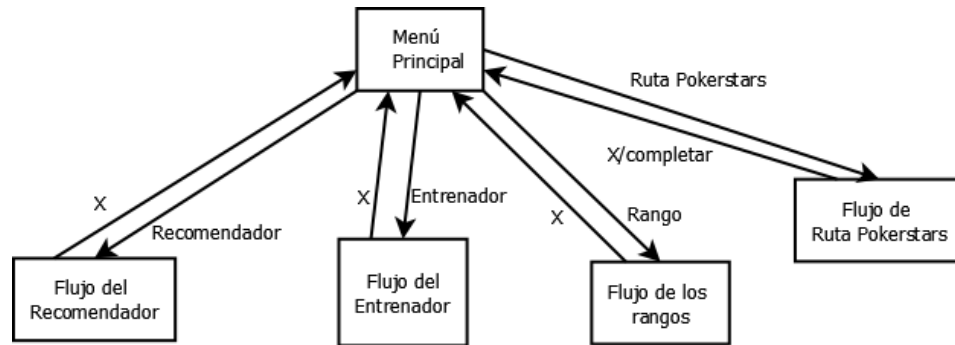


Figura 4.1: Diagrama de flujo desde el menú principal

la encuentre, esperará el turno del jugador, mostrándole la recomendación mientras sea el turno, una vez se pase, esperará al siguiente turno del jugador, como se observa en la figura 4.2.

4.1.2. Flujo del Entrenador

Una vez estemos en la ventana del entrenador, podremos seleccionar que tipo de jugada queremos generar con los botones o generar una propia manualmente. Después seleccionaremos la acción a realizar (FOLD, RAISE, CALL), y evaluaremos la jugada. A continuación se mostrará la información de si hemos acertado o no y podremos generar una nueva jugada.

4.1.3. Flujo de la herramienta de rangos

En esta ventana podremos o bien crear nuevos rangos o bien modificar algunos de los ya establecidos. Para crear un nuevo rango seleccionamos la situación en la que estaremos (OR, 3-Bet, etc) la posición (SB,BB,MP, etc) y finalmente las jugadas para ese rango. Guardaremos el rango dándole el nombre que queramos.

Para modificar un rango existente lo seleccionaremos de la lista desplegable de rangos, de la tabla seleccionaremos las jugadas que queramos añadir o quitar, modificando el rango y pulsaremos en reescribir. Podremos dejarle el mismo nombre o modificarlo y después guardar.

4.1.4. Ruta de PokerStars

Se nos abre una ventana para seleccionar la ubicación, podemos o bien seleccionar la ubicación y aceptar o cancelar.

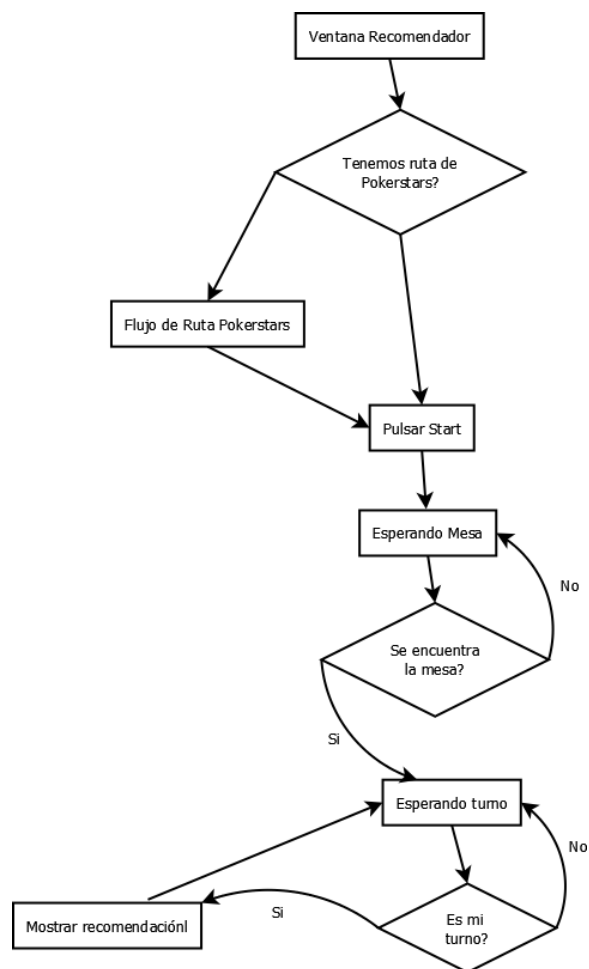


Figura 4.2: Diagrama de flujo del recomendador

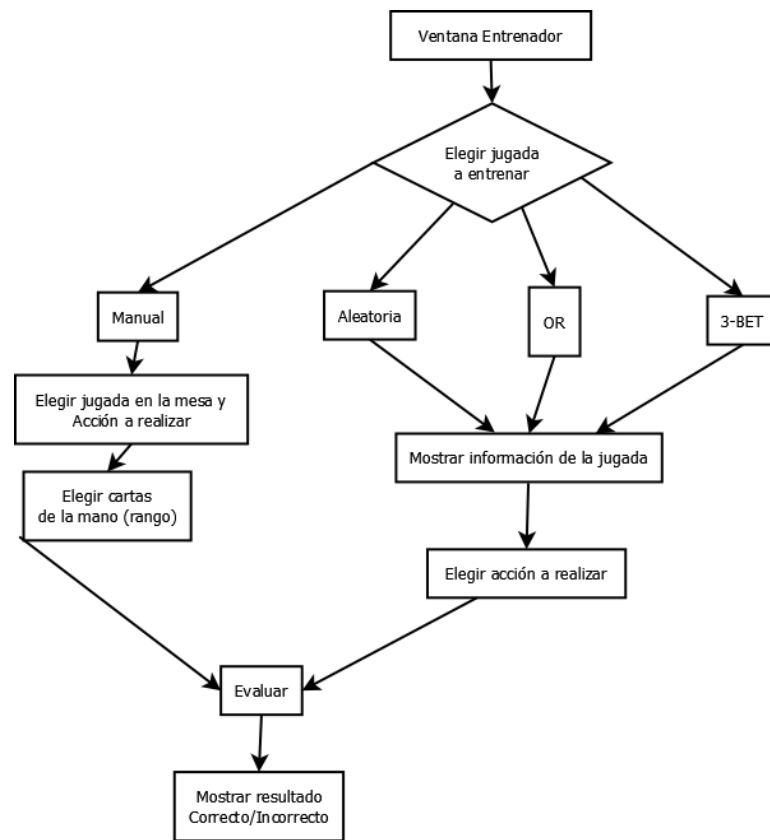


Figura 4.3: Diagrama de flujo del entrenador

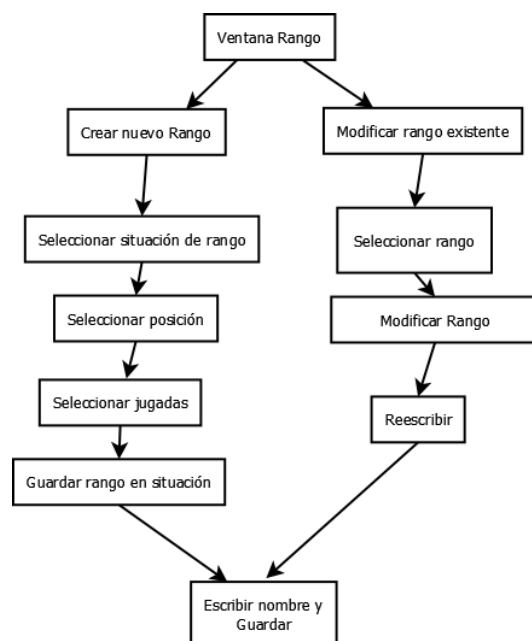


Figura 4.4: Diagrama de flujo del cambio de los rangos

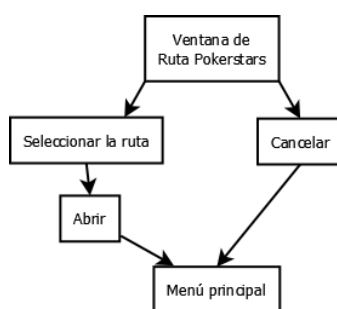


Figura 4.5: Diagrama de flujo del seleccionador de ruta de PokerStars

4.2. Uso del programa

El recomendador y entrenador Pre-Flop para NLHE **Senshei Poker** ofrece al usuario una oportunidad para aprender y mejorar su juego.

Para ello, al arrancar el programa se ofrecen al usuario cuatro opciones:

- **Recomendador:** Recomienda al usuario qué jugada será la más adecuada teniendo en cuenta el estado actual de la mesa de juego y los rangos considerados.
- **Entrenador:** Ayuda al usuario a practicar jugadas. A su vez está dividido en tres opciones: aleatorio, 3-bet y Open Range.
- **Rangos:** Permite al usuario modificar los rangos establecidos o añadir los suyos propios.
- **Ruta PokerStars:** Permite al usuario modificar la ruta donde se guardan las manos de PokerStars.



Figura 4.6: Aspecto del menú principal

Cómo se observa en la figura 4.6, el usuario puede elegir cuál de estas funcionalidades desea utilizar de cara a mejorar su juego.

4.2.1. Recomendador

El recomendador se encargará de sugerir al usuario la mejor jugada en el turno Pre-Flop exclusivamente, aunque el juego continúe después en las siguientes fases.

Para utilizarlo el usuario debe tan solo tener PokerStars en primer plano, en inglés, y con una partida en juego, y haber habilitado en los ajustes de PokerStars la opción de “Save hands”.

Una vez se seleccione la opción de Recomendador, nos aparecerá la ventana correspondiente. Si no tenemos la ruta guardada para las manos de Pokerstars nos saldrá una pantalla pidiéndonos la ruta. Tras ello nos encontraremos en la ventana del recomendador. Esta contiene dos botones: start y stop. Para comenzar a usarlo pulsaremos en start, y el programa esperará a encontrar la ventana con el tablero de juego (figura 4.7).



Figura 4.7: Aspecto del recomendador mientras espera a encontrar la mesa

El recomendador esperará al turno del usuario (figuras 4.8 y 4.9) y tras leer la información sobre la mesa y calcular las jugadas posibles, ofrece al usuario la mejor opción con la información disponible, es decir, la que genere el mayor beneficio (o la menor pérdida) al jugador (figura 4.10).

Después de esto, el usuario podrá llevar a cabo la jugada recomendada o aquella que considere más oportuna. Una vez el usuario decida y actúe, el recomendador esperará a nuestro siguiente turno Pre-Flop.

En cualquier momento, el usuario podrá parar el recomendador pulsando “Stop” o cerrando la ventana, con lo que regresará al menú principal.



Figura 4.8: Aspecto del recomendador mientras espera al turno del usuario



Figura 4.9: Aspecto del recomendador cuando llega el turno del usuario



Figura 4.10: Aspecto del recomendador tras calcular una recomendación

4.2.2. Entrenador

Esta funcionalidad permite al usuario poner en práctica jugadas para diversas situaciones de juego para entrenar qué jugada realizar en un momento específico. El entrenador permitirá al usuario elegir la jugada y después le comunicará si ha elegido la jugada óptima. Para ello se han creado tres tipos de entrenador

- **Aleatorio:** El entrenador ofrece al usuario una situación aleatoria.
- **Open Raise:** El usuario debe realizar la apertura de la mesa, según su posición.
- **3-Bet:** Una jugada en la que el usuario se enfrentará a la posibilidad de realizar una tercera subida.

Al acceder a la funcionalidad del entrenador, al usuario se le muestran varias partes con las que interactuar, como se muestra en la figura 4.11.

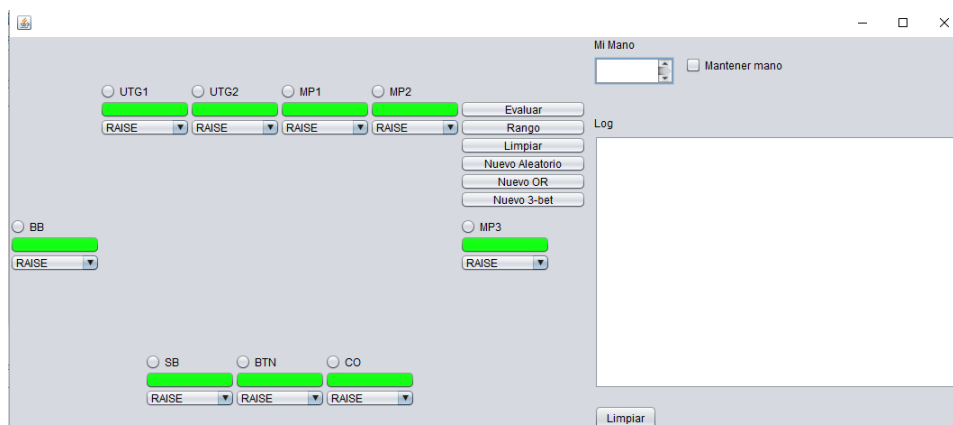


Figura 4.11: Aspecto del entrenador

En primer lugar, se observa a la izquierda una situación de las posiciones de una mesa de póker. Justo al lado, hay una serie de botones con los que el usuario puede interactuar:

- **Evaluar:** Se pulsará tras decidir qué hacer en una jugada generada e introducir el movimiento.
- **Limpiar:** Limpia la mesa de las opciones tomadas, dejándola como nueva para una nueva jugada. También limpia la ventana de logs.
- **Nuevo Aleatorio:** Genera una nueva jugada aleatoria.
- **Nuevo OR:** Genera una nueva jugada de apertura.

- **Nuevo 3-Bet:** Genera una nueva jugada de 3-Bet.

Por otro lado, hay dos ventanas de información en el lado derecho de la ventana. La superior indica la jugada que el usuario tiene en la mano y se podrá modificar con el botón de Rangos o dejarla como está cuando generemos una jugada. La inferior indica de forma sencilla cuál es la jugada. Una vez el usuario decida qué hacer, el entrador indicará si la decisión ha sido un acierto o un error.

Para empezar a entrenar el usuario debe elegir qué tipo de jugada desea generar: Aleatoria, Open Range o 3-Bet, y pulsar sobre el botón de la jugada elegida. Ésta se mostrará en la parte de la izquierda, y la sección de log mostrará la información de la que se dispone (figura 4.12).

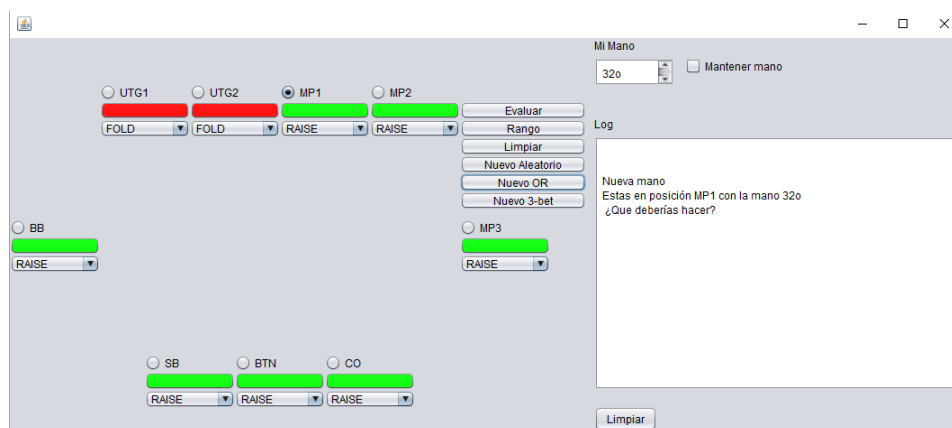


Figura 4.12: Entrenador tras generar una jugada Open Range

La posición del usuario en la jugada generada se indica mediante el círculo seleccionado en la parte izquierda, así como en la información del log. También dispone de los datos de los jugadores que se encuentran antes del jugador, así como las cartas que tiene en la mano, mostradas en la ventana de “Mi Mano”. Con esta información, el usuario deberá elegir qué hacer, seleccionando una de las opciones del desplegable correspondiente a su posición (CALL, RAISE o FOLD).

Una vez seleccionado un movimiento, el usuario pulsará “Evaluar” para saber si la decisión ha sido adecuada o no, tras lo cual el programa mostrará en la ventana de log la siguiente información (figura 4.13):

- **Situación:** Rango que se ha usado para evaluar la jugada.
- **Su acción/Acción ideal:** Muestra el movimiento elegido y la acción óptima.

- **Resultado:** Muestra de forma visual si se ha acertado o no en la jugada.

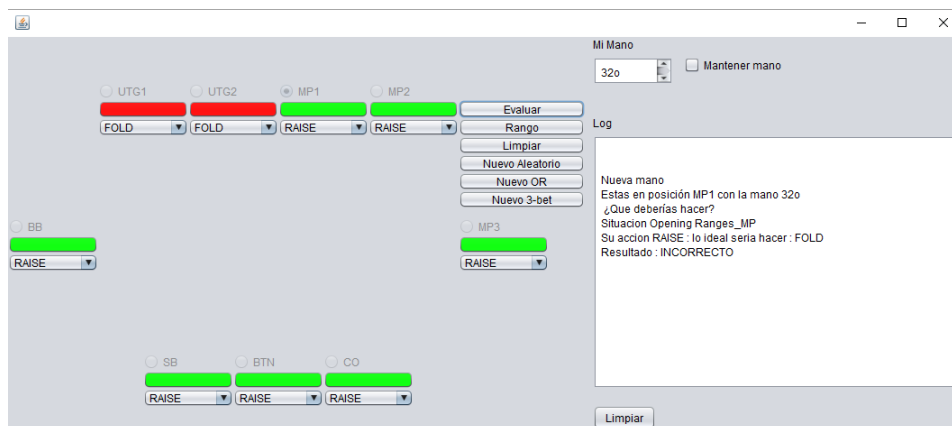


Figura 4.13: Entrenador tras evaluar la opción elegida

Ahora el usuario podrá generar una nueva mano con los botones de “Nuevo Aleatorio”, “Nuevo OR” o “Nuevo 3-Bet” para continuar entrenando. Si no se desea tener toda la información de entrenamiento de la sesión hasta ahora, se puede borrar el log con el botón “Limpiar”, disponible debajo del mismo.

También podrá limpiar la mesa anterior para generar una jugada propia. Si el usuario intenta generar una jugada imposible, el entrenador mostrará un mensaje indicando que no es una mano reconocible en el póker. Por ello se recomienda no utilizar esta opción a no ser que se trate de un jugador de póker de nivel medio, sugiriendo a los usuarios novatos que utilicen las jugadas generadas con las opciones automáticas.

4.2.3. Rangos

En esta ventana el usuario podrá ver, modificar o añadir nuevos rangos para utilizar con nuestro recomendador. Para ello se presenta la interfaz de la figura 4.14.

Si el usuario desea crear un rango nuevo, deberá seguir los siguientes pasos:

1. En el desplegable “Situación” de la parte derecha, seleccionará la situación para la que desea generar el rango.
2. Justo debajo, seleccionará la posición en la que se encuentra.

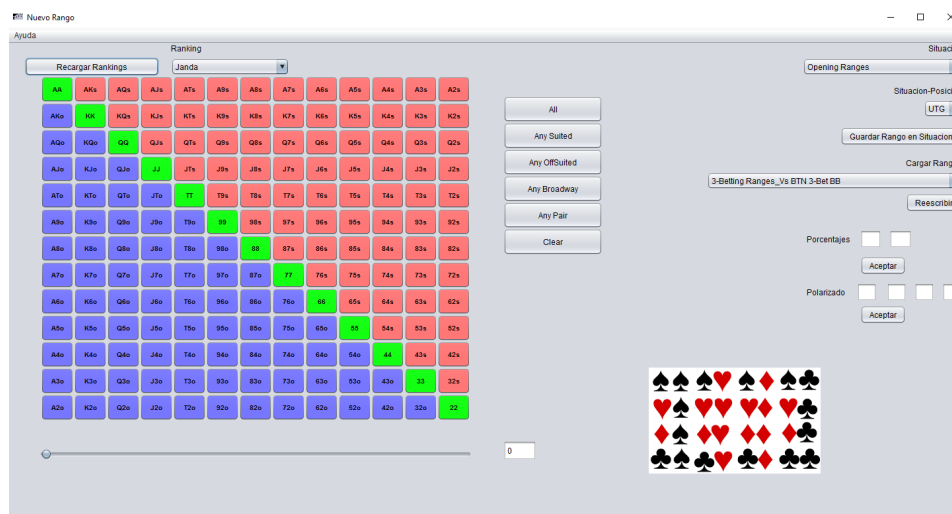


Figura 4.14: Aspecto de la ventana de Rangos

3. Seleccionar sobre la tabla de la izquierda las jugadas que queramos en nuestro rango.
4. Pulsará sobre “Guardar Rangos en situación”, dará un nombre y guardará.

Si por otro lado el usuario desea modificar un rango existente, deberá seleccionar en “Cargar rangos” a la derecha el rango que desea modificar (figura 4.16).

En la tabla de la izquierda aparecen en amarillo las jugadas que componen el rango seleccionado. Una vez cargado, el usuario tiene varias opciones para modificarlo:

- **Jugadas:** Quitar o poner a mano las jugadas que quiere o no quiere incluir, pulsando sobre ellas en la tabla.
- **Porcentajes:** Coger el porcentaje de manos a partir del primer valor y hasta el segundo valor.
- **Polarizar:** Tomar los valores polarizados entre los porcentajes establecidos para añadir valores de “farol”. Esta modificación se plantea para usuarios de nivel alto de conocimiento de póker.

El usuario podrá usar una de estas opciones o cualquier combinación de la primera con alguna de las otras dos.

Hecho esto, el usuario pulsará sobre “Reescribir rangos”, dará un nombre o sobrescribirá uno existente y guardará.

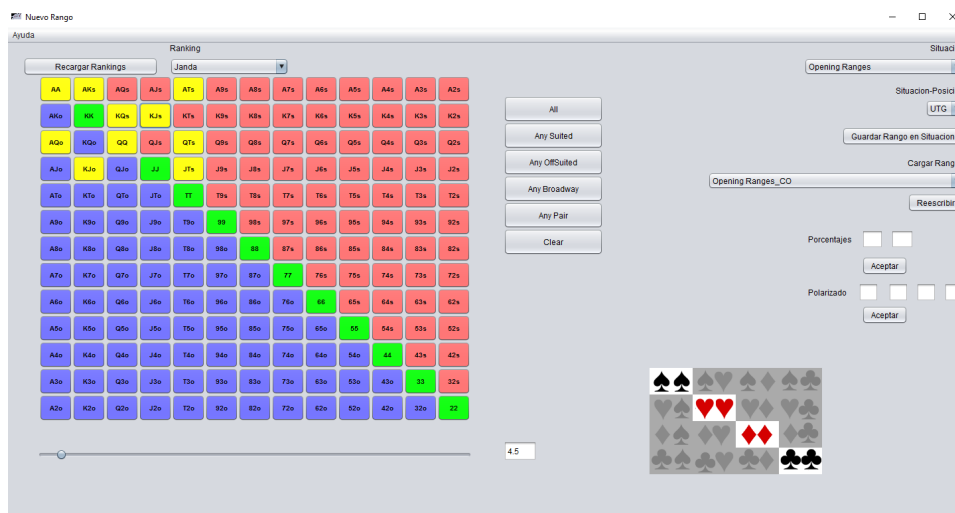


Figura 4.15: Ejemplo de creación de un Open Range en UTG



Figura 4.16: Ejemplo de modificación del Opening Ranges_CO



Figura 4.17: Ejemplo de modificación de Opening Ranges_CO, seleccionando entre el 10 % y el 50 % y modificando jugadas

4.2.4. Ruta de PokerStars

Al seleccionar la opción de Ruta de PokerStars, se abre una ventana (figura 4.18) donde el usuario podrá seleccionar la ruta donde desea guardar las manos de PokerStars. El programa utilizará estas manos para adaptar los rangos cuando se tenga suficiente información de los jugadores.

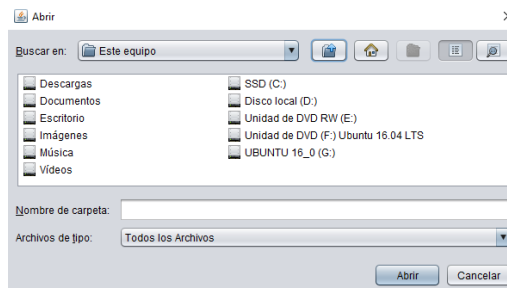


Figura 4.18: Ventana de selección de ruta

4.3. Resultados

En este proyecto se construyó un producto software que no genera una salida a partir de una entrada, si no que supone un soporte para gente que está empezando en el mundo del póker online. Por ello, en vez de presentar los resultados de las pruebas unitarias realizadas a lo largo del proceso software, parecería más relevante aportar los resultados de un test de usabilidad, para estudiar si los usuarios encuentran la herramienta útil e intuitiva de cara a mejorar su juego.

El test de usabilidad que se presentó (anexo A) consiste en tres partes, cuyos resultados se analizan de forma anónima. Las partes del test de usabilidad son las siguientes:

- **Formulario de control:** Breve cuestionario anónimo de clasificación de los participantes por edad, género, estudios, ocupación y conocimiento sobre póker.
- **Tareas:** Una serie de tareas a realizar con la herramienta, durante las cuales se pide a los participantes que piensen en voz alta y hagan comentarios y sugerencias durante el proceso. Asimismo se miden los tiempos y errores durante la realización de cada tarea usando la hoja de mediciones A.
- **Cuestionario final:** Un cuestionario de evaluación de la aplicación y una nueva oportunidad para que el participante exprese su opinión.

El test de usabilidad se realizó con tres participantes, de los cuales uno se consideraba un jugador novel de póker, uno decía tener un nivel medio y el tercero era experto. Todos eran hombres y mayores de edad, aunque tenían distintos niveles de estudios y ocupaciones.

Las tareas, que se pueden consultar en el anexo A, tomaron una media de 54.3 minutos a cada participante, con la primera siendo la más costosa, tanto en tiempo como en errores, seguida de la tarea 3.2, mientras que la más sencilla fue la tarea 4. Estos datos se pueden comprobar en las gráficas de las figuras 4.19 y 4.20.

En general, los participantes encontraron que la aplicación cumple con su cometido: ofrece una ventaja a la hora de utilizar el recomendador durante el juego, además de una herramienta útil de aprendizaje. Sin embargo, encontraron que algunos aspectos de la aplicación resultaban complicados al uso y podrían tener una interfaz mejorada. También se sugirieron algunas ampliaciones o mejoras, como incluir una explicación breve de los conceptos básicos para los usuarios más principiantes o ampliar la funcionalidad a otras



Figura 4.19: Tiempo medio empleado en cada tarea



Figura 4.20: Media de errores por tarea

fases de la mano, más allá del Pre-Flop. Como se observa en la gráfica de la figura 4.21, las puntuaciones medias son relativamente satisfactorias, con una puntuación mínima de 3 para todos los criterios.

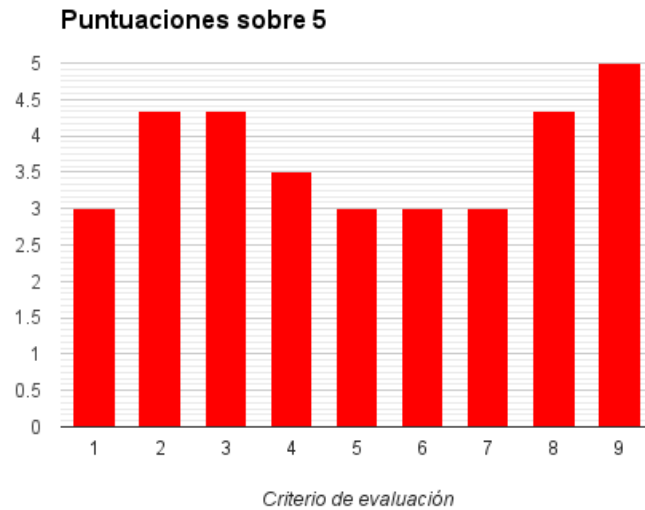


Figura 4.21: Puntuación media de satisfacción según los criterios de evaluación del cuestionario final

La parte de la aplicación que más gustó y más útil resultó fue el recomendador, a pesar de que encontraron incómodos los tiempos de espera. Por otro lado, los rangos resultaron menos agradables, por ser considerada la sección más compleja y que más requería de explicación de conceptos.

Capítulo 5

Conclusiones y líneas futuras

*No olvides que siempre habrá otra
partida de poker, si no mañana, pasado
mañana o la semana que viene. Me
gusta pensar en el poker como un juego
constante que dura toda mi carrera.*

Jennifer Harman

RESUMEN: Tras finalizar el proceso de desarrollo de un proyecto software, es primordial ver hasta dónde hemos llegado y qué hemos conseguido. En este capítulo vamos a ver qué objetivos hemos cumplido, los problemas que hemos encontrado por el camino y las posibles formas de avanzar en el proyecto en un futuro.

5.1. Conclusiones

En este proyecto hemos conseguido cumplir todos los objetivos planteados al inicio del proceso, es decir, construir tanto un recomendador como un entrenador para la fase Pre-Flop de juegos de póker NLHE online en la sala PokerStars. Más específicamente hemos desarrollado:

- Un reconocedor de imagen eficiente de la situación de la mesa de juego en un momento dado.
- Un recomendador de jugadas adaptativo en función de nuestros rivales y de como hayan jugado hasta entonces.
- Un entrenador para mejorar y aprender a hacer frente a las jugadas más comunes en el póker.

Además, como equipo hemos cumplido una serie de objetivos personales de cara a la realización del proyecto, como es aprender a gestionar un equipo, el uso de la metodología ágil Scrum, aunque fuera en una versión reducida, el uso de \LaTeX como herramienta de generación de documentos, y mejorar nuestro propio juego de póker.

Hemos creado una herramienta que puede ser muy útil para aquellos jugadores recreacionales de póker que quieran mejorar sus resultados online. A pesar de que el recomendador sólo funcione para la fase Pre-Flop, esta fase supone la mayoría del tiempo de juego y aprender a jugar bien aquí puede suponer una mejora también en las demás fases. También puede servir para que jugadores no profesionales pero que jueguen de forma habitual puedan entrenar y perfeccionar su juego.

Nuestra conclusión al final de este proyecto es que cualquiera puede llegar a jugar decentemente al póker, con tan sólo un entrenamiento, tiempo, y ganas de aprender. Los resultados del test de usabilidad confirman que nosotros hemos facilitado este proceso y reducido la lectura que el usuario hubiera tenido que realizar.

5.2. Problemas encontrados en el desarrollo y sus soluciones

Todo proyecto software conlleva un proceso costoso para los integrantes del equipo y, por tanto, se presentan situaciones problemáticas que deben ser resueltas para su correcta finalización.

Uno de los problemas con el que nos encontramos durante el proceso de desarrollo fue el retraso en algunas de las entregas planificadas en los objetivos iniciales. Esto se produjo porque al iniciar el proyecto se pensó que las labores de investigación y la implementación de ciertas características tomarían menos tiempo. Otro factor importante en este problema fue la falta de tiempo de uno de los miembros de equipo, cuya dedicación a su empleo a tiempo completo dejó menos tiempo del necesario para este proyecto, así como las demás asignaturas y prácticas de laboratorio pendientes. Este problema se solucionó cuando propuso una planificación mejorada con estimaciones de tiempos para los Sprints más acordes al tiempo disponible de cada miembro del equipo.

A nivel técnico, uno de los mayores problemas encontrados fue que en un primer momento se decidió usar la pantalla de tamaño predeterminado de PokerStars, pero ésta depende de la resolución del monitor, por lo que

en cuanto no se usaba el tamaño adecuado nuestro programa fallaba. La solución fue hacer el reconocedor escalable, lo que conllevó a su vez una investigación acerca de los tamaños y formas de obtener la información.

Otro obstáculo fue cómo utilizar la base de datos de PokerTracker para obtener los datos. Inicialmente, se pensaba que sería un proceso sencillo, pero resultó ser más complicado de lo esperado dada la gran envergadura de los datos. Al final se decidió postponer la inclusión de la base de datos hasta después de concluir el recomendador básico, cuando se pudiera dedicar más tiempo a trabajar con la base de datos.

Hubo otro rompecabezas notable: la localización del tablero de juego. En un principio se quiso utilizar la imagen del logotipo de PokerStars como método para encontrar la ventana de juego, pero tras lo costoso de la implementación, resultó que este método no sería válido, puesto que la posición del logo respecto a lo demás es dependiente de la versión de Windows, por lo que hubo que buscar una forma alternativa de localizar el tablero de juego.

5.3. Líneas futuras

Al tratarse de un trabajo de fin de grado, el proceso se vió limitado por la falta de tiempo y recursos. Con una mayor dedicación, el producto podría llegar a ser mucho más funcional, añadiendo recomendaciones para el resto de las fases de juego, FLOP, TURN y RIVER; aunque para el caso de NLHE con más de dos jugadores en mesa, no se dispone todavía de un algoritmo que calcule la solución óptima.

Otra forma de continuar el proyecto podría ser que, en vez de recomendar, el programa jugase automáticamente. Esta ampliación probablemente no le hiciera mucha gracia a las casas de juego, siendo de hecho, ilegal y estando explícitamente prohibida en los “Terms of use”. Sin embargo, podría ampliarse la funcionalidad actual a otras salas de juego, como pueden ser 888Poker, TitanBet, etc.

En cualquier caso, la primera línea de actuación debería consistir en aplicar algunas de las sugerencias de los participantes del test de usabilidad, puesto que al ser una herramienta enfocada al usuario, hacer que la interfaz de usuario sea lo más usable posible es prioritario.

Finalmente, se podrían eliminar las restricciones actuales sobre las opciones de PokerStars, es decir, el número de jugadores en mesa, el tipo de las cartas, la opción de diseño de interfaz, el idioma, el sistema operativo utilizado, etc.

A pesar de ser una plataforma relativamente reducida actualmente, tiene mucho potencial de crecimiento, y aún así supone para los jugadores recreacionales una gran ventaja respecto a otros que no dispongan de nuestra herramienta.

Capítulo 6

Conclusions and future lines of work

Don't forget that there will always be another poker game, if not tomorrow, the day after tomorrow or next week. I like to think of poker as a constant game that lasts my whole career.

Jennifer Harman

RESUMEN: After finishing the development process of a software project, it is vital to recap what we've achieved. In this chapter we will see which objectives we accomplished, the problems we encountered along the way and the possible ways to continue the project in the future.

6.1. Conclusions

In this project we have accomplished all the objectives that were set at the beginning of the process; we built a recommender and a trainer for the Pre-Flop phase for NLHE online poker in the Pokerstars site. More specifically we have developed:

- An image recogniser for the situation in the board in the moment of our turn.
- An adaptative recommender depending on our rivals and how they have played.
- A trainer to improve and learn how to play against the most common hands in poker.

Furthermore, as a team we accomplished a series of personal objectives in this project, like managing a team, the use of the agile methodology Scrum, even if in a reduced version, the use of L^AT_EX as a document generation tool, and improving our own poker play.

We have created a tool that can be very useful to those recreational poker players that want to improve their online results. Although the recommender only works for the Pre-Flop phase, this phase is the phase that is played the most of the time, and learning how to play it could be a great improvement for the next phases. It can also be useful to those non-professional players who play regularly and want to train and perfect their play.

Our conclusion at the end of this project is that everyone can play poker decently, with only a bit of training, time and motivation to learn. The results of the usability test confirm that we have made this process easier and reduced the reading that the player would have had to do.

6.2. Problems found in development and solutions

Every software project is an expensive process for the integrants of the team and, for that, some problematic situations come up that must be solved for the project's completion.

One of the problems that we faced during the development process was the delay in some of the deadlines planned in the inicial objectives. This was due to a miscalculation of the time needed for investigation and implementation of certain aspects. Another factor in this issue was the tight schedule of one of the team members, whose dedication to his full time job allowed for less time than necessary for the project, as well as other classes and lab projects. The issue was solved when an improved plan was presented with time estimations for each Sprint that were more accurate to each team member's availability.

On the technical level, one of the bigger problems found was that we initially decided to use the default size of the PokerStars window, but it depends of the resolution of the monitor, so when the size wasn't appropriate, it wouldn't work. The solution was to make the recommender scalable, which meant studying the different sizes and ways to obtain the information.

Another obstacle was how to use the PokerTracker database to obtain the necessary data. Initially, we thought that it would be an easy process, but it was complicated due to the database structure and the size of the data. Finally, we decided to postpone the inclusion of the database to after

finishing the basic recommender, when we had more time.

And there was another notorious puzzle: the localization of the poker table. On a first attempt we used the PokerStars logo as the method to find the game window, but after an expensive implementation, this method turned out to not be valid, because the position of the logo respect other things in the window depends on the Windows version, so we had to look for another way to localize the game window.

6.3. Future lines

Because this is a Graduate Thesis, the process was limited by the lack of time and resources. With a greater dedication, the product could be more functional, adding recommendations for the next phases of the game, FLOP, TURN and RIVER; though for the case of NLHE with more than two players at the table, we do not yet have an algorithm to calculate the optimal solution.

Another way to continue the project could be that, instead of recommending an action, the program played automatically. This ampliation would probably not be looked upon kindly by online poker sites, being, in fact, illegal and explicitly forbidden in the “Terms of use”. Nevertheless, the current functionality could be applied to other game sites, like 888Poker, Titanbet, etc.

In any case, the first future line of work must be to apply some of the suggestions from the participants of the usability test, because this is a user focused tool, and so making a user interface that is friendly and as usable as possible is prioritary.

Finally, we could delete the restrictions about the options in Pokerstars, that is, the number of players at the table, the theme of the cards, the design of the user interface, language, Operative System used, etc.

Despite of currently being a somewhat reduced plataform, it has lot of growt potential, and even now gives recreational players a great advantage over others that do not have this tool.

Apéndice A

Test de usabilidad

Introducción

Estamos evaluando esta herramienta realizando con usted un test de usabilidad. Durante el test le agradeceremos que se sienta libre de realizar cualquier comentario, sugerencia o pregunta que se le ocurra. El test de usabilidad consiste en la realización de una serie de tareas con la aplicación, que le explicaremos a continuación. Tenga en cuenta que estamos evaluando la aplicación, no a usted, por lo que no va a hacerlo ni bien ni mal, nuestro objetivo es ver el uso de la aplicación en una situación real para identificar problemas y fallos que presente.

El primer paso será un pequeño formulario general de clasificación. Todos los datos serán anónimos, no tendrá que darnos ninguna información identificativa. Continuaremos con la realización de las tareas indicadas, durante la cual tomaremos anotaciones que corresponderán con el tiempo empleado y los comentarios que vaya haciendo. Finalmente procederemos con un pequeño cuestionario para recoger información de su satisfacción en el uso del producto.

¡Muchas gracias por su colaboración!

Formulario de control

- Género
 - Hombre
 - Mujer
 - Otros (indique cual)
- Edad
 - <18
 - 18-24
 - 25-29
 - 30-39
 - 40-49
 - 50-59
 - 60-69
 - 70+
- Ocupación
 - Estudiante
 - Informático
 - Otros (indique cual)
- Nivel de estudios
 - Educación secundaria obligatoria (ESO)
 - Bachiller o COU
 - Formación Profesional (FP)
 - Estudios Universitarios (licenciatura o grado)
 - Estudios de post-grado (máster o doctorado)
 - Otros (indique cual)
- Conocimiento sobre póker
 - Novel
 - Medio
 - Alto

Tareas a realizar

Las tareas que debe realizar son las siguientes:

1. Recomendador

Juegue 10 de manos en PokerStars, aplicando las recomendaciones del programa.

- a) ¿Le parece clara la indicación de las esperas entre manos?
- b) ¿Le parece clara la recomendación?

2. Entrenador

a) Generar una jugada aleatoria

Genere una jugada aleatoria pulsando el botón correspondiente y posteriormente seleccionando la jugada a realizar y después el botón evaluar;

- 1) ¿Le parece claro el botón de seleccionar la jugada?
- 2) ¿Es capaz de reconocer la situación de la mesa?
- 3) ¿Es capaz de encontrar su posición?
- 4) ¿Es capaz de seleccionar su jugada?
- 5) ¿El botón de Evaluar le parece claro?
- 6) ¿Le parece claro como limpiar la ventana de Log?
- 7) ¿Le parece claro como reiniciar/limpiar la mesa?

b) Generar jugada Open Raise

Genere una jugada Open Raise pulsando el botón correspondiente y posteriormente seleccionando la jugada a realizar y después el botón evaluar;

- 1) ¿Le parece claro el botón de seleccionar la jugada?
- 2) ¿Es capaz de reconocer la situación de la mesa?
- 3) ¿Es capaz de encontrar su posición?
- 4) ¿Es capaz de seleccionar su jugada?
- 5) ¿El botón de Evaluar le parece claro?
- 6) ¿Le parece claro como limpiar la ventana de Log?
- 7) ¿Le parece claro como reiniciar/limpiar la mesa?

c) Generar jugada 3-Bet

Genere una jugada de 3-Bet pulsando el botón correspondiente y posteriormente seleccionando la jugada a realizar y después el botón evaluar;

- 1) ¿Le parece claro el botón de seleccionar la jugada?
- 2) ¿Es capaz de reconocer la situación de la mesa?
- 3) ¿Es capaz de encontrar su posición?

- 4) ¿Es capaz de seleccionar su jugada?
- 5) ¿El botón de Evaluar le parece claro?
- 6) ¿Le parece claro como limpiar la ventana de Log?
- 7) ¿Le parece claro como reiniciar/limpiar la mesa?

3. **Rangos** Solo debe realizar estas tareas si se considera un usuario medio o avanzado.

a) **Crear un nuevo rango**

Cree un nuevo rango desde 0 seleccionando las jugadas que desee de la tabla interactiva de la izquierda, seleccione el tipo de rango que es, su posición en la mesa y guarde.

- 1) ¿Le parece sencillo seleccionar las jugadas?
- 2) ¿Y la situación?
- 3) ¿Y la posición?

b) **Modificar un rango existente**

Modifique uno de los rangos existentes, seleccionandolo con el desplegable y cambiando las jugadas con la tabla interactiva de la parte izquierda, y/o usando las opciones de porcentajes o polarizando.

- 1) ¿Le parece sencillo seleccionar el rango?
- 2) ¿Y modificarlo con la tabla interactiva?
- 3) ¿Y modificarlo con los porcentajes/polarizarlo?

4. **Cambiar la ruta del fichero de Pokerstars**

Modifique la ruta actual de las manos guardadas a un directorio distinto.

- a) ¿Le parece sencillo el cambio?

Cuestionario final

Por favor evalúe los siguientes criterios del 1 al 5, siendo 1 la puntuación mínima y 5 la máxima:

1. Facilidad de uso de la aplicación en general.
2. Utilidad del recomendador.
3. Utilidad del entrenador.
4. Utilidad de modificar los rangos (si aplica).
5. Utilidad de cambiar la ruta.
6. Estructura y organización de la aplicación.
7. Representatividad y claridad de los botones y ventanas de la aplicación.
8. En general, ¿le fue intuitivo usar el recomendador?
9. Teniendo en cuenta el propósito de la aplicación, ¿le parece que cumple su cometido?

Por favor, conteste brevemente a las siguientes preguntas:

10. ¿Qué parte le ha gustado más? ¿Por qué?
11. ¿Cuál le ha gustado menos? ¿A qué es debido?
12. ¿Qué sección considera la más complicada? ¿Cuál podría ser su causa?
13. Por favor, comparta su opinión personal y cualquier observación acerca de la aplicación:

Medidas del test

A continuación se muestran las medidas que se tomarán durante la realización del test:

- Tiempo total:
- Total de errores:
- Medidas de tarea, se podrán anotar en la tabla A.1:
 - (A) Tiempo empleado en realizar la tarea
 - (B) Número de errores
 - (C) Tiempo empleado en errores y recuperación
 - (D) Frecuencia en la que se ha necesitado ayuda

Tabla A.1: Medidas de usabilidad para las tareas

	A	B	C	D
Tarea 1				
Tarea 2 a)				
Tarea 2 b)				
Tarea 2 c)				
Tarea 3 a)				
Tarea 3 b)				
Tarea 4				

Bibliografía

*Intenta decidir lo buena que es tu mano
en un momento determinado. No
importa nada más. ¡Nada en absoluto!*

Doyle Brunson

GÓMEZ-MARTÍN, M. A. y GÓMEZ-MARTÍN, P. P. *TEX_S: Una Plantilla de L^AT_EX Para Tesis y otros documentos, Manual de Usuario*. 2009. Disponible en <http://gaia.fdi.ucm.es/files/nightlybuilds/TeX_S/TeX_S-Manual-NightlyBuild.pdf> (último acceso, Junio 2016).

JANDA, M. *Applications of No-Limit Poker*. Two Plus Two Publishing LLC, 2013. ISBN 1-880-68555-8.

MAX-VALUE-SOFTWARE. *PokerTracker*. 2016. Disponible en <https://www.pokertracker.com/> (último acceso, Junio 2016).

MUÑOZ, S. *Rankings de Sklansky-Chubukov*. 2009. Disponible en <http://www.poker-red.com/articulos/estrategia/rankings-sklansky-chubukov> (último acceso, Junio 2016).

NGUYEN, Q. *Tess4J, A Java JNA wrapper for Tesseract OCR API*. 2016. Disponible en <http://tess4j.sourceforge.net/> (último acceso, Junio 2016).

OETIKER, T., PARTL, H., HYNÄ, I. y SCHLEGL, E. *The Not So Short Introduction to L^AT_EX 2_ε*. Versión electrónica, 1996.

OSKARI TAMMELIN, M. J., NEIL BURCH y BOWLING, M. Solving heads-up limit texas hold'em. 2015. Disponible en <http://poker.cs.ualberta.ca/publications/2015-ijcai-cfrplus.pdf> (último acceso, Junio 2016).

POKERSTARS.ES. *Poker Texas Hold'em*. 2016a. Disponible en <https://www.pokerstars.es/poker/games/texas-holdem/> (último acceso, Junio 2016).

POKERSTARS.ES. *PokerStars*. 2016b. Disponible en <https://www.pokerstars.es/> (último acceso, Junio 2016).

AUTORIZACIÓN PARA LA DIFUSIÓN DEL TRABAJO FIN DE GRADO Y SU DEPÓSITO EN EL REPOSITORIO INSTITUCIONAL E-PRINTS COMPLUTENSE

Los abajo firmantes, alumno/s y tutor/es del Trabajo Fin de Grado (TFG) en el Grado en Ingeniería en Informática.....de la Facultad de Informática....., autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el Trabajo Fin de Grado (TF) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Periodo de embargo (opcional):

☐ 6 meses

☐ 12meses

TÍTULO del TFG:Recomendador adaptativo y entrenador pre-flop para NLHE.....

Curso académico: 20.....¹⁵ / 20.....¹⁶

Nombre del Alumno/s:

Juan Carlos Marco González

Iván Martín Herrero

Tutor/es del TFG y departamento al que pertenece:

Manuel Nuñez García

Departamento de Sistemas Informáticos y Computación

Firma del alumno/s



Firma del tutor/es

